Treball de Fi de Grau

Grau en Enginyeria de Tecnologies Industrials

Aplicació dels models de regressió logística i k-Nearest Neighbors a la predicció de resultats acadèmics

MEMÒRIA

Autor: Carla Vidal Montesinos

Director: Lluís José Talavera

Convocatòria: Febrer 2020

Escola Tècnica Superior

d'Enginyeria Industrial de Barcelona

RESUM

El present document tracta sobre l'anàlisi del rendiment de tècniques de mineria de dades aplicades en la predicció de l'aprovat o suspès dels estudiants de l'ETSEIB en assignatures corresponents al Q3. La mineria de dades és el procés d'extracció d'informació significativa d'un conjunt de dades mitjançant la identificació de patrons i tendències. Les tècniques de predicció emprades són la regressió logística i l'algorisme *K-Nearest Neighbors*.

El procés d'anàlisi està adaptat a la metodologia CRISP DM que comprèn les diferents etapes que cal seguir per poder aplicar un model de mineria de dades. A partir dels resultats obtinguts s'han contrastat els dos models de predicció utilitzats.

El software utilitzat durant el treball ha estat *Python* i s'ha fet ús de les seves llibreries *Pandas*, *scikit-learn* i la distribució *Anaconda* com a IDE.

La conclusió principal que s'extreu del treball és que, en general, *K-Nearest Neighbors* és millor que la regressió logística. Tanmateix, hi ha assignatures on els resultats no són satisfactoris, probablement degut a la distribució de classes desequilibrada. A la secció final es proposen alternatives per tal de millorar l'anàlisi.

INDEX

1. INTRODUCCIÓ 5
1.1. Objectius 7
1.2. Abast del projecte 8
1.3. Eines utilitzades 9
2. COMPRENSIÓ I PREPARACIÓ DE DADES 11
3. MODELATGE I VALIDACIÓ 18
3.1. Tècniques de mineria de dades: 18
3.2. Regressió logística 19
3.3. K-Nearest neighbors 24
4. VALIDACIÓ 39
4.1. Mètodes de validació 39
4.2. Mètriques de rendiment 40
4.3. Predicció mitjançant Regressió Logística 44
4.4. Predicció mitjançant K-Nearest Neighbors 49
4.5. Comparació entre mètodes predictius 64
5. PRESSUPOST 67
6. IMPACTE AMBIENTAL 69
7. CONLUSIONS 70
BIBLIOGRAFIA 72
ANNEX 73

1. INTRODUCCIÓ

El continu avanç de la tecnologia per a la gestió de bases de dades, així com la digitalització de molts processos, ha fet possible integrar imatges, vídeos, textos i altres dades numèriques en una base de dades senzilla.

La mineria de dades neix com a conseqüència de la generació massiva de dades, les quals són analitzades mitjançant tècniques automàtiques amb l'ajuda dels ordinadors o mètodes que involucren l'acció humana amb l'objectiu de descobrir patrons i tendències. És un subcamp de la informàtica que combina tres disciplines científiques: l'estadística , la intel·ligència artificial i l'aprenentatge automàtic (algorismes que poden aprendre de les dades per fer prediccions).

Requereix la identificació d'un problema, juntament amb la recollida de dades que poden conduir a una millor comprensió i models informàtics per proporcionar anàlisis estadístics o altres mitjans. El procés d'explotació de dades depèn de la tecnologia de la informació, la forma d'emmagatzematge i del software per analitzar les dades. No obstant, el procés de la mineria de dades requereix de la feina per part de l'analista en la selecció del model, selecció i transformació de dades i interpretació del resultats. La seva aplicació apareix en diferents àrees així com el comerç, la banca o la sanitat.

Un processament actual de la nostra vida quotidiana és l'enregistrament de productes mitjançant codis de barres, el qual ha fet molt còmode la compra i proporciona grans masses de dades als establiments. Les botigues poden processar ràpidament les nostres compres i utilitzar els ordinadors per a determinar amb precisió el preu dels productes i ajudar en la gestió d'inventaris. Juntament amb altres fonts d'informació, la informació recopilada mitjançant el codi de barres es pot utilitzar per a l'anàlisi de mineria de dades. Un anàlisi adequat de les dades pot donar lloc a nova informació, així com identificar aquells subconjunts de clients més rendibles pel negoci i enfocar els productes a les seves necessitats.

L'exemple clàssic de l'aplicació de la mineria de dades és la detecció d'hàbits de compra al supermercat. Un estudi molt citat va detectar que els divendres hi havia una quantitat inusualment elevada de clients que adquirien alhora bolquers i cervesa. Profunditzant més en l'estudi, es va descobrir que aquell dia els pares joves anaven al supermercat a comprar bolquers i, de pas, compraven una cervesa pel cap de setmana. El supermercat va poder incrementar les seves vendes de cervesa col·locant-les properes als bolquers per fomentar les ventes compulsives.

També ha estat molt utilitzada en l'àmbit mèdic, on permet fer diagnòstics més precisos. Disposar d'una base de dades que conté informació del pacient així com registres mèdics, exàmens físics i pautes de tractament, permet receptar tractaments més efectius. També permet una gestió més eficaç, eficient i rendible dels recursos sanitaris mitjançant la identificació de riscos, la predicció de malalties en determinats segments de la població o la previsió de la durada d'ingrés a l'hospital.

Finalment, una de les aplicacions amb èxit de la mineria de dades per la qual va obtenir un gran reconeixement va ser la detecció de fraus de targetes de crèdit. Els algorismes d'aprenentatge automàtic podien detectar o predir el frau a través de patrons inusuals de les dades recopilades.

En resum, la mineria de dades ha demostrat ser una eina extremadament útil que aplica la tecnologia en la recerca de patrons ocults i relacions interessants en grans volums de dades.

Metodologia CRISP

Per tal de realitzar l'anàlisi de la mineria de dades de manera sistemàtica, es segueix una metodologia. Hi ha diferents processos estandarditzats, entre els quals destaca la metodologia CRISP. El procés CRISP-DM (*Cross-Industry Standard Process for Data Mining*) és un procés estàndard àmpliament utilitzat pels membres de la indústria. Aquest model consta de sis fases dins un procés cíclic començant amb l'exploració de dades, la recollida de dades, el processament de dades, anàlisi, inferències extretes i implementació.

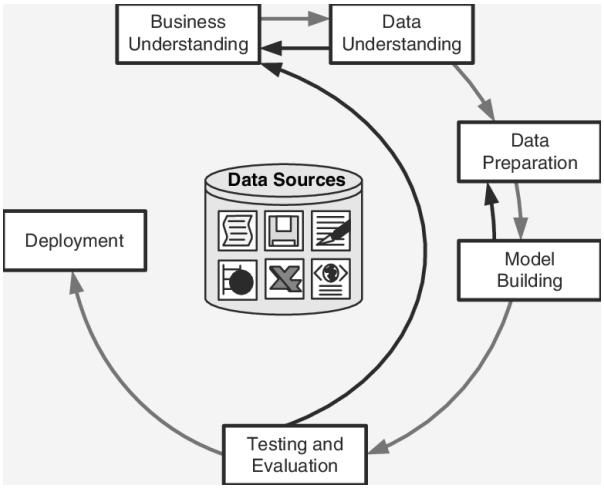


Figura 1. Procés CRISP DM ***[1]***

1. ***Comprensió del problema:*** Inclou determinar els objectius de l'estudi, l'avaluació de la situació actual i el desenvolupament d'un pla de projecte. Aquest coneixement permet preparar les dades i interpretar els resultats de forma correcta.

2. ***Comprensió de les dades:*** Un cop establerts els objectius i pla del projecte, es du a terme la recol·lecció inicial de dades i la seva familiarització. Aquest pas inclou la recollida, descripció, exploració i la verificació de la qualitat de les dades. És necessària una bona selecció de dades disponibles per a descriure correctament una determinada tasca. Hi ha un mínim de tres qüestions a considerar en la selecció de dades. La primera és establir una descripció concisa i clara de la documentació del problema. La segona qüestió és identificar les dades rellevants per a la descripció del problema. Per últim, és imprescindible que les variables seleccionades per a les dades rellevants siguin independents les unes amb les altres. La independència de les variables significa que no existeix informació sobreposada.

3. ***Preparació de les dades:*** Un cop determinades les dades disponibles cal seleccionar-les, netejar-les i transformar-les en la forma desitjada per tal de preparar-les per les tècniques de mineria de dades que s'utilitzaran posteriorment. En general, la neteja de dades significa filtrar, agregar i emplenar els valors buits. Mitjançant un filtratge de les dades, es detecten l'existència d'*outliers* i redundàncies. Els *outliers* són aquells valors

destacats que es troben fora del rang dels grups de dades seleccionats. Poden sobresortir per moltes raons, com ara errors humans o tècnics o de manera natural a causa d'esdeveniments extrems. Les dades redundats són aquelles que contenen la mateixa informació registrada en diverses formes. Existeixen molts mètodes estadístics i eines de visualització que serveix per preprocessar les dades seleccionades. Estadístiques comunes com el màxim, el mínim, la mitjana i el mode es poden utilitzar fàcilment per agregar o suavitzar les dades. Els *Box plots* i *Scatter plots* es solen utilitzar per filtrar els *outliers*. La transformació de dades consisteix en l'aplicació de fórmules matemàtiques o corbes d'aprenentatge per a convertir diferents mesures de dades en un conjunt unificat d'escala numèrica. Es pot utilitzar la transformació de les dades per eliminar les diferències d'escales en les variables, per a reduir o ampliar les dades donades, o per a recodificar les dades categòriques a dades numèriques. En alguns casos, el preparació de dades podria assumir el 50% del temps procés de mineria de dades.

4. ***Modelatge:*** En aquesta fase es seleccionen les tècniques de modelatge més adequades segons el tipus de dades i els objectius establerts. Un cop establertes, es procedeix a la construcció del model. També s'estableix el pla de prova, un procediment destinat a provar la qualitat i la validesa del model. Habitualment es divideixen les dades en dos subconjunts: d'entrenament (*Training set)* i de proves (*Testing set)* de manera que es construeix el model en base al conjunt d'entrenament i es mesura la qualitat d'aquest mitjançant el conjunt de prova.

5. ***Validació i avaluació:*** En aquesta etapa s'avaluen els resultats del model mitjançant mètriques de rendiment dels models. En el cas que els resultats no compleixin amb els criteris d'èxit establerts, serà necessari revisar el procés i repetir algun pas anterior.

6. ***Implementació:*** Una vegada s'ha construït i validat el model, el coneixement obtingut s'ha de transformar en accions que comportin la millora de l'àmbit pel projecte. És recomanable guardar els resultats obtinguts del projecte, de manera que es disposi de proves documentades per a futurs estudis. També és aconsellable preparar estratègies de monitorització i manteniment per tal que el coneixement obtingut estigui controlat per si sorgeixen canvis.

1.1. Objectius

L'objectiu d'aquest treball és realitzar un estudi del rendiment de les tècniques de Regressió Logística i *k-Nearest Neighbours* aplicades a la predicció de resultats acadèmics dels estudiants del grau d'enginyeria industrial a l'ETSEIB *(Escola Tècnica Superior d'Enginyeria Industrial de Barcelona)* de les assignatures corresponents al primer quadrimestre de la fase no inicial (Q3). Per dur a terme aquesta tasca s'aplicarà de forma rigorosa una metodologia pel desenvolupament de projectes de mineria de dades i s'utilitzaran les llibreries *Pandas* i *sk-learn* de *Python*.

Els objectius principals són:

- **Aplicació d'una metodologia.** Per a realitzar un anàlisi de dades s'aplicarà de forma rigorosa una metodologia ben documentada amb la identificació clara de les seves fases, de manera que pugui ser replicada en un futurs anàlisis.

- **Aplicació de diferents conjunts de dades.** S'estudiaran tres conjunts de dades amb diferents variables i es compararan els resultats obtinguts.

- **Estudi i comparació de tècniques de mineria de dades.** S'aplicaran les tècniques descrites anteriorment i a partir dels resultats obtinguts, s'analitzarà el rendiment dels models en funció d'uns certs paràmetres i es contrastaran els algorismes.

Es poden definir els objectius secundaris:

- **Familiarització amb l'entorn.** La programació del codi es durà a terme amb l'IDE *Anaconda.* Anaconda incorpora totes les eines necessàries així com la llibreria *Scikit-learn* que proporciona diversos algorismes per l'anàlisi de dades i la instal·lació d'editors com *Spyder*.

- **Implementació amb la llibreria *Pandas*.** Es treballarà amb la llibreria *Pandas* de Python per a la manipulació de les dades durant la fase de preparació. És necessari un cert grau de coneixement d'aquesta per tal d'implementar funcions al llarg del codi informàtic.

1.2. Abast del projecte

El present treball tracta d'un projecte d'anàlisi de mineria de dades, el qual seguirà la metodologia CRISP MD explicada anteriorment. Les seves fases s'han adaptat als objectius i limitacions del treball, desglossant-lo en quatre apartats principals:

1. **Comprensió del problema**. La part d'introducció exposa els objectius i requisits del treball. Es considera que, com a estudiant, el coneixement del problema que es planteja és suficient, ja que s'han cursat totes les assignatures del grau amb les seves diferents metodologies i avaluacions.

2. **Comprensió i preparació de les dades.** La comprensió de les dades no requereix d'una gran dedicació, doncs les variables inicials no són complexes. A partir de la descripció de les dades i les seves variables, es procedeix a preparar-les. La fase de preparació consta de dues etapes: la neteja i selecció de dades i la seva posterior transformació. S'organitzaran les dades amb l'estructura desitjada mitjançant un codi informàtic de manera que la preparació sigui automàtica.

3. **Modelatge.** Una vegada s'han estructurat les dades, es procedirà a la construcció del model. Es seleccionen els mètodes de predicció més adequats i s'estudia la seva efectivitat en funció dels seus possibles paràmetres.

4. **Avaluació i resultats.** Un cop construït el model s'avaluen els resultats mitjançant tècniques de validació i mètriques de rendiment.

L'abast del projecte comprèn la majoria de les fases d'un projecte de mineria de dades convencional. No s'inclou la fase d'implementació a causa de trobar-se limitada; es tractaria de posar en funcionament el mètode de predicció a l'escola.

1.3. Eines utilitzades

Les eines utilitzades durant les fases de l'anàlisi de dades seran informàtiques ja que es tracta d'un estudi de mineria de dades. Les eines emprades són les següents:

Python

Figura 2. Python

Python és un llenguatge de programació d'alt nivell amb una semàntica dinàmica integrada principalment per al desenvolupament de webs i aplicacions.

És un llenguatge relativament senzill, per la qual cosa és fàcil d'aprendre ja que la seva sintaxi està centrada en la llegibilitat. Els desenvolupadors poden llegir i traduir codi *Python* molt més fàcil que altres llenguatges.

Python s'imparteix a les dues assignatures d'informàtica de l'escola, *Fonaments d'Informàtica* i *Informàtica,* raó principal per la qual s'ha escollit aquest tipus de llenguatge a la implementació del codi.

Un dels avantatges més prometedors de *Python* és que totes les eines necessàries estan disponibles per a tothom de forma gratuïta. Aquestes eines inclouen una gran quantitat de llibreries per diferents àrees d'estudi, algunes de les quals seran d'utilitat per a la realització d'aquest treball.

Pandas

Pandas és una llibreria de *Python* de codi obert que s'utilitza per la manipulació de dades d'alt nivell. Incorpora una sèrie de funcions que s'utilitzaran per la part de neteja i transformació de dades. Alguns punts a destacar d'aquesta llibreria són:

- Disposa d'eines de lectura i escriptura de dades de formats típics com fiters .cvs i .txt, documents de càlcul com .xslx i bases de dades com SQL.
- Treball amb l'objecte *Dataframe* propi d'aquesta llibreria basat en una taula que incorpora automàticament la indexació de files.
- Addició o transformació de dades mitjançant la operació *groupby,* de manera que s'apliquen diferents funcions a conjunts de dades específics.

Anaconda

Figura 3. Anaconda

Anaconda és la plataforma estàndard de *Python* formada per un gran nombre de llibreries i entorns de treball de *Data Science.* La instal·lació del paquet *Anaconda* inclou tots els elements de treball d'entre els quals es farà ús de l'editor *Spyder* i la llibreria *Scikit-learn.*

- ***Spyder***
  El codi programat durant el treball s'editarà en el programari *Spyder,* un *IDE*[1] desenvolupat en *Python* format per una aplicació informàtica que proporciona un entorn de treball en el desenvolupament del *Software.*

  Conté un editor amb diferents eines perquè l'edició del codi sigui òptima, una consola que permet executar comandes i un explorador de variables, el qual mostra tots els elements creats en la consola durant la programació, així com variables, mòduls o funcions.

- ***Scikit-learn***
  Scikit-learn és una llibreria *Python* que ve instal·lada amb la distribució *Anaconda.* Proporciona algorismes per l'anàlisi de dades, de manera que l'usuari estalvia tems en la programació del codi durant la construcció del model i li permet enfocar-se més en el propi anàlisi. Alguns d'ells són algorismes de classificació, regressió i *clustering.*

Figura 4. Scikit-learn

[1] *Integrated Development Environment*

2. COMPRENSIÓ I PREPARACIÓ DE DADES

La comprensió i preparació de les dades és la base de l'anàlisi posterior. És un procediment clau en el resultat de l'estudi: la presa de decisions en la selecció, rebuig o transformació de dades a explorar pot canviar considerablement els resultats. És per això que sol ser una de les parts més llargues i elaborades en quant a la presa de decisions.

2.1. Descripció de les dades inicials:

Es disposa de les dades inicials dels resultats acadèmics dels alumnes de l'ETSEIB del període comprés entre 2010 i 2017. Aquestes dades es divideixen en tres documents *Excel*: dades de preinscripció, dades de la fase inicial i dades de la fase no inicial.

Cada fitxer conté diferents variables disposades en columnes on cada fila correspon a les dades acadèmiques de la matrícula d'un estudiant per una assignatura concreta. Cada vegada que es matricula una assignatura s'afegeix una fila. En cas que una assignatura es repeteixi, la informació no es sobreescriu, s'emplenen dues files en la base de dades. Així doncs, cada matrícula d'una assignatura ocupa una fila de la taula de dades.

***Dades de preinscripció*.** El document conté informació dels alumnes en forma de les següents variables:

Variable	Descripció
CODI_EXPEDIENT	Codi assignat a l'expedient de cada alumne per a la identificació d'aquest.
SEXE	Home (H) o Dona (D).
CP_FAMILIAR	Codi postal de la residència familiar de l'alumne.
ANY_ACCES	Any d'accés a la universitat.
TIPUS_ACCES	Tipus d'accés. Hi ha un únic valor 1 assignat.
NOTA_ACCES	Nota de selectivitat de l'alumne amb la que accedeix al grau.
CP_CENTRE_SEC	Codi postal del centre d'educació secundària d'on procedeix l'alumne.

Taula 1. Descripció de l'arxiu de les dades de preinscripció

***Dades de la Fase inicial i Fase no inicial*.** El document conté informació sobre les assignatures cursades pels alumnes en forma de les següents variables:

Variable	Descripció
CODI_PROGRAMA	Codi del grau o màster cursat. En el cas dels estudiants del GETI (*Grau en Enginyeria en Tecnologies Industrials*) és el 752.
CODI_EXPEDIENT	Codi assignat a l'expedient de cada alumne.
CODI_UPC	Codi de l'assignatura matriculada corresponent a la qualificació.
CURS	Any en que l'alumne es matricula de l'assignatura.
CREDITS	Nombre de crèdits ECTS de l'assignatura.
QUAD	Quadrimestre en què es cursa l'assignatura. Q1 correspon al quadrimestre de tardor i Q2 al de primavera.

SUPERA	Indica si es supera l'assignatura. "S" si l'assignatura està aprovada (*Sí supera*) i "N" si suspesa (*No supera*).
NOTA_PROF	Nota de l'assignatura assignada pel professor.
NOTA_NUM_AVAL	Nota de l'assignatura per a l'avaluació curricular.
NOTA_NUM_DEF	Nota final definitiva passat el període d'avaluació curricular.
GRUP_CLASSE	Grup de classe de l'assignatura en què l'alumne està matriculat.

Taula 2. Descripció dels arxius de les dades de la fase inicial i no inicial

2.2. Preparació de les dades:

Tal i com s'ha mencionat abans, les dades inicial es troben en tres taules de fitxers *excel*. Aquestes taules s'han convertit al format *DataFrame* per tal de poder treballar amb la llibreria *Pandas.*

Es durà a terme tres procediments diferents amb un seguit de transformacions per tal de fusionar les dades, on el resultat final serà la obtenció de tres *DataFrames*. El que es desitja és organitzar la informació de diverses formes a través de diferents variables en els *DataFrames* i comparar els resultats obtinguts.

El primer *DataFrame* utilitzarà com a variables les qualificacions dels alumnes de la fase inicial i la fase no inicial. El segon i tercer *DataFrame* contindran la mateixa informació que el primer incorporant una variable addicional en cadascun d'ells, la nota de selectivitat en el segon i el nombre de repeticions de les assignatures en el tercer.

Es seguirà el següent procediment: la neteja i selecció de dades, comuna a tots els *DataFrames*, i la transformació de dades, diferent per a cadascun dels *DataFrames* finals. Tot el codi emprat en la preparació de dades es troba a l'Annex.

1) Neteja i selecció de dades

Les dades inicials contenen una gran quantitat de dades mentre que aquest estudi es limita només a un cert domini . L'estudi només s'aplicarà als estudiants que segueixin unes certes condicions. Es seleccionen les dades corresponents als estudiants del GETI, els quals prenen el valor de *752* en la variable *CODI_PROGRAMA*.

A continuació, es selecciona únicament els alumnes que hagin superat la fase inicial, doncs només aquest grup seran necessaris per a la predicció. S'afegeix una columna anomenada *Superaini* la qual agafa la última nota de l'assignatura de la columna *'SUPERA'* . A continuació, es filtren els estudiants que tinguin el valor *S* a la columna *Superaini.*

L'estudi només considerarà les assignatures del primer quadrimestre de la fase no inicial corresponents al segon curs. Aquestes assignatures es seleccionaran mitjançant el seu *CODI_UPC* .

S'ordenen cronològicament les dades segons el curs i el quadrimestre que es cursa l'assignatura. S'eliminen les dades corresponents al quadrimestre zero[2] ja que només es tindran en compte els quadrimestres de tardor i primavera.

[2] "*Quadrimestre zero o curs d'introducció: Alguns centres ofereixen a les estudiantes i estudiants de nou accés als estudis de grau la possibilitat de realitzar un curs d'introducció en el quadrimestre de tardor, de manera que l'inici dels estudis oficials s'ajorna fins al quadrimestre de primavera."*

Els *Missing values* són aquelles cel·les que no contenen cap valor. Una vegada es converteixen a *DataFrame* prenen el valor de *Nan* (*Not a number).* Es filtren els estudiants que tenen un valor buit a la seva nota de la columna. Es pot trobar el cas d'algorismes que no acceptes variables d'entrada de tipus *Nan*, de manera que se'ls hi haurà de donar un valor o, com en el nostre cas, eliminar-los.

2) Transformació de les dades

L'objectiu d'aquesta fase és obtenir tres matrius de dades, anomenades també *DataFrames* segons la llibreria *Pandas,* que permetin analitzar les qualificacions obtingudes pels estudiants. La informació necessària per a cadascun d'ells serà la nota obtinguda per a cada estudiant, de manera que s'obviarà la informació referent als crèdits, curs, quadrimestre i grup de classe.

A continuació es descriuran els passos seguits per a l'obtenció de cadascun dels tres *DataFrames* finals i les seves respectives variables. El procediment és similar per a les tres propostes de taules i segueix el següent esquema: obtenció d'una primera taula que varia segons el *Dataframe* que finalment s'unirà amb una segona taula comuna per a tots.

Per a l'obtenció de la primera taula s'utilitzaran les dades dels fitxers Excel de la fase inicial i de les dades de preinscripció (aquest últim només pel cas del *DataFrame 2*).

Dataframe 1

La matriu de dades mostrarà el *CODI_EXPEDIENT* de cada estudiant en cadascuna de les files i les columnes seran la nota mitjana obtinguda en cadascuna de les assignatures de la fase inicial.

CODI_EXPEDIENT	ALG	CALC1	...	QUIM2	EXPRE
Estudiant A					
Estudiant B					
Estudiant C					
Estudiant D					

Taula 3. Dataframe 1

1. Crear una nova columna amb la nota mitjana de l'assignatura per a cada fila. Per a crear aquesta columna s'utilitza l'operació *groupby* de la llibreria *Pandas* de manera que s'agrupen les dades de cada assignatura i estudiant. Això permet calcular la mitjana de les notes amb la funció *mean* i afegir la columna *NOTA_MITJ* amb els resultats. Mitjançant la funció *drop_duplicates* s'eliminen les files amb la mateixa assignatura, nota mitjana i *CODI_EXPEDIENT,* quedant-nos amb l'últim duplicat. A continuació, s'elimina la columna *NOTA_DEF* ja que només ens interessa la mitjana.

CODI_EXPEDIENT	CODI ASSIG	NOTA DEF	NOTA MITJ
Estudiant A	ALG	3	4,5
Estudiant A	ALG	6	4,5
Estudiant B	QUIM1	5	5
Estudiant B	EXPRE	4	6
Estudiant B	EXPRE	8	6

CODI_EXPEDIENT	CODI ASSIG	NOTA DEF	NOTA MITJ
Estudiant A	ALG	6	4,5
Estudiant B	QUIM1	5	5
Estudiant B	EXPRE	8	6
Estudiant C	ALG	6	6
...			

Figura 5. Procés pel càlcul de la nota mitjana

2. Tal i com s'ha explicat abans, les dades inicials contenen per a cada fila una convocatòria cursada per a una determinada assignatura i expedient. Aquest format no és útil pel modelatge de dades. Es desitja transformar les dades en una nova taula on les dades d'un mateix expedient es trobin en una sola fila. El procés emprat rep el nom de *pivoting,* de manera que les assignatures passen a ser columnes de la taula i com a valors la nota mitjana de les assignatures. La llibreria *Pandas* conté una funció *Pandas.Dataframe.pivot* que transforma les dades inicials a un format com el descrit. La taula final conté una fila per expedient, una columna per a cada assignatura i les cel·les contenen les notes mitjanes.

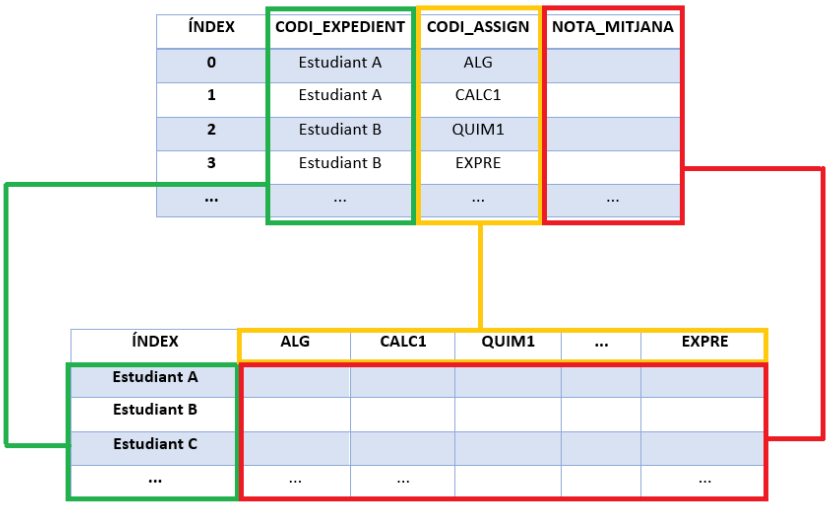


Figura 6. Procés de pivoting

Dataframe 2

La matriu de dades mostrarà la mateixa informació que el *Dataframe 1* afegint com a nova columna la nota de selectivitat.

CODI_EXPEDIENT	ALG	CALC1	...	QUIM2	EXPRE	SELE
Estudiant A						
Estudiant B						
Estudiant C						
...						

Taula 4. Dataframe 2

1. Crear una nova columna amb la nota mitjana de les assignatures.

2. S'afegirà la nota de selectivitat, *NOTA_ACCES,* a la columna de les qualificacions *NOTA_MITJANA* a partir de l'Excel amb les dades de preinscripció dels alumnes. Es canviarà el nom de la columna *NOTA_MITJANA* a *Notadef.* Cada fila que contingui una nota de selectivitat prendrà el valor *Sele* per a la columna *CODI_ASSIGN.* La funció *pandas.DataFrame.append* de la llibreria *Pandas* permet dur a terme aquesta acció.

3. Es normalitzarà la columna *Notadef* donat que els seus valors presenten escales diferents. La nota mitjana de les assignatures del grau pren valors d'entre 0 i 10 mentre que la nota de selectivitat presenta un rang de 0 a 14. Aquesta última influirà més en el resultat intrínsecament degut als seus valors més grans. L'objectiu de la normalització és canviar els valors de les columnes numèriques del conjunt de dades a una escala comuna, sense distorsionar les diferències en els intervals de valors. Es portarà el conjunt de dades a una mateixa escala, de manera que els elements de la columna *Notadef* prendran valors d'entre 0 i 1. D'aquesta manera, el model serà menys sensible a les diferències d'escala presentant millors resultats. Aquesta acció es duu a terme mitjançant la funció *sklearn.preprocessing.MinMaxScaler()* la qual transforma les variables a un rang imposat, en el nostre cas (0,1).

4. Finalment caldrà pivotar els valors de la mateixa manera que pel *DataFrame 1,* les assignatures seran les columnes de la nova taula i les notes normalitzades els valors de les cel·les.

Dataframe 3

La matriu de dades mostrarà la mateixa informació que el *Dataframe 1* afegint com a nova columna el nombre de repeticions per a cada assignatura.

CODI_EXPEDIENT	ALG	R_ALG	CALC1	R_CALC1	...	QUIM2	R_QUIM2
Estudiant A							
Estudiant B							
Estudiant C							
...							

Taula 5. Dataframe 3

1. Crear una nova columna amb la nota mitjana de les assignatures.

2. Crear una nova columna *REP* amb el nombre de repeticions. Per fer-ho s'utilitza altre cop l'operació *groupby* de la llibreria *Pandas* de manera que es fan grups de les dades de cada assignatura i estudiant. La funció *cumcount* numera els diferents elements que hi ha a cada grup per ordre. Les dades venen ordenades cronològicament de la fase de selecció i neteja de dades, de manera que l'última fila de cada grup serà el nombre de repeticions total. Mitjançant la funció *drop_duplicates* s'eliminen les files amb la mateixa assignatura, nota mitjana i *CODI_EXPEDIENT,* quedant-nos amb l'última fila duplicada que contindrà el nombre de repeticions total. Els resultats obtinguts d'aplicar la funció *cumcount* es troben a la columna *REP.* Finalment, s'elimina la columna *NOTA_DEF*.

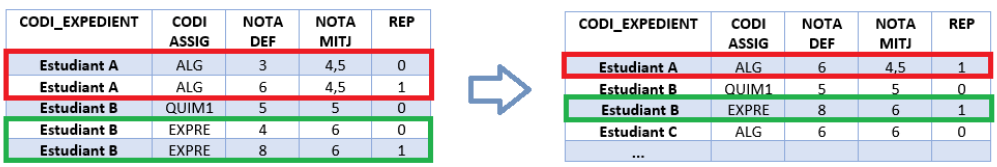
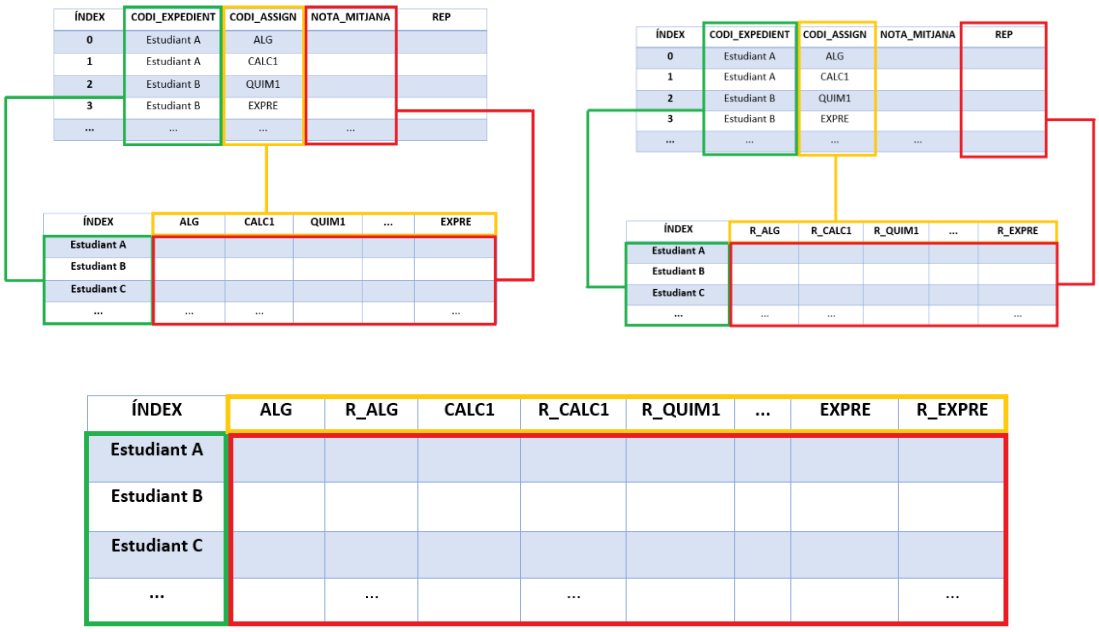


Figura 7. Procés de càlcul del nombre de repeticions

3. Es normalitzaran les columnes *Notamitj* i *Rep* donat que els seus valors presenten escales diferents. La nota mitjana de les assignatures del grau pren valors d'entre 0 i 10 mentre que el nombre de repeticions presenta un rang de 0 a 4 .Es portarà el conjunt de dades a una mateixa escala on els elements de la columna *Notadef* i la columna *Rep* prendran valors d'entre 0 i 1.

4. Es pivotaran els valors de manera que el nombre de repeticions i les assignatures siguin les columnes de la nova taula. Per fer-ho, es pivotaran dues taules prenent com a valors les notes mitjanes i el nombre de repeticions respectivament. Finalment, s'ajuntaran les dues taules aplicant la funció *Pandas.DataFrame.merge*.



ÍNDEX	ALG	R_ALG	CALC1	R_CALC1	R_QUIM1	...	EXPRE	R_EXPRE
Estudiant A								
Estudiant B								
Estudiant C								
...		...		...				...

Figures 8, 9 i 10. Procés de pivoting per les dues taules i la seva fusió posterior

La segona taula correspon als valors objectius de l'estudi (y), és a dir, les notes que es volen predir. Per a l'obtenció d'aquesta s'utilitzaran les dades de l'Excel de la fase no inicial. La matriu de dades mostrarà el *CODI_EXPEDIENT* de cada estudiant en cadascuna de les files i les columnes seran la nota mitjana de les sis assignatures del primer quadrimestre de la fase no inicial en format binari.

CODI_EXPEDIENT	INFO	MEC	EDOS	MATERS	ELECTRO	METNUM
Estudiant A						
Estudiant B						
Estudiant C						
...						

Taula 6. Segon taula corresponent a les assignatures del Q3

El procediment seguit per a l'obtenció d'aquesta és el següent:

1. Crear una nova columna amb la primera nota de l'assignatura del Q3. A partir de les notes del la fase inicial, es vol predir què treuen els alumnes la primera vegada que cursen una assignatura del Q3. El procediment és el mateix que per calcular la nota mitjana però ara s'utilitza la funció *first* en lloc de *mean.*

2. Tenint present els futurs algorismes de predicció que s'utilitzaran, s'haurà de tenir en compte el format de dades de les variables. Les tècniques que s'utilitzaran són la Regressió Logística *k-Nearest Neighbors* aplicades a la predicció de resultats acadèmics. Aquestes s'utilitzen principalment per a problemes de classificació. En concret, aquest estudi es tracta d'un problema de classificació binària, ja que s'haurà de predir les notes dels estudiants com a aprovades 'A' o suspeses 'S'. És per això que es requereix que la variable dependent sigui binària. Cadascuna de les assignatures de la fase no inicial prendrà el paper de variable de sortida. Convertirem les variables numèriques de les notes mitjanes en variables categòriques binàries. Si la nota mitjana és superior o igual a 5 prendrà el valor d'aprovat 'A' , altrament se li assignarà la lletra 'S'.

3. Es pivotaran la matriu de dades amb l'eina *pivot.* Les assignatures passaran a ser les columnes i les notes mitjanes en format binari els valors de les cel·les.

Una vegada obtingudes les dues taules, caldrà fusionar-les mitjançat la funció *Pandas.DataFrame.merge*. Es descarten aquells expedients que presenten un valor *Nan* en algun valor de les columnes del *DataFrame.* Per últim, caldrà modificar la nomenclatura. Les assignatures venen representades per un codi numèric donat per l'escola. Es vol substituir el codi de l'assignatura per el seu codi alfabètic per tal que sigui més fàcil d'interpretar visualment. Les assignatures amb el seu codi numèric i alfabètic corresponent es mostra a continuació:

Codi numèric	Assignatura	Codi alfanumèric
240011	Àlgebra Lineal	ALG
240012	Càlcul I	CALC1
240013	Mecànica Fonamental	MECFON
240014	Química I	QUIM1
240015	Fonaments d'informàtica	FONINFO
240021	Geometria	GEO
240022	Càlcul II	CALC2
240023	Termodinàmica Fonamental	TERMO
240024	Química II	QUIM2
240025	Expressió Gràfica	EXPRE
240132	Informàtica	INFO
240133	Mecànica	MEC
240131	Equacions diferencials	EDOS
240033	Materials	MATERS
240031	Electromagnetisme	ELECTRO
240032	Mètodes Numèrics	METNUM

Taula 7. Codi numèric i alfanumèric corresponent a les assignatures

3. MODELATGE I VALIDACIÓ

Un cop estructurades les dades en la forma desitjada, es procedeix al seu anàlisi. L'anàlisi comprèn dues fases: la construcció del model i la seva validació. Aquestes es duen a terme alhora i comporten un procés cíclic: la construcció del model inicial implica una primera validació i a partir dels resultats obtinguts el seu ajustament posterior.

3.1. Tècniques de mineria de dades:

Dins el camp de l'aprenentatge automàtic[3], hi ha dos tipus principals de tècniques: les supervisades i les no supervisades.

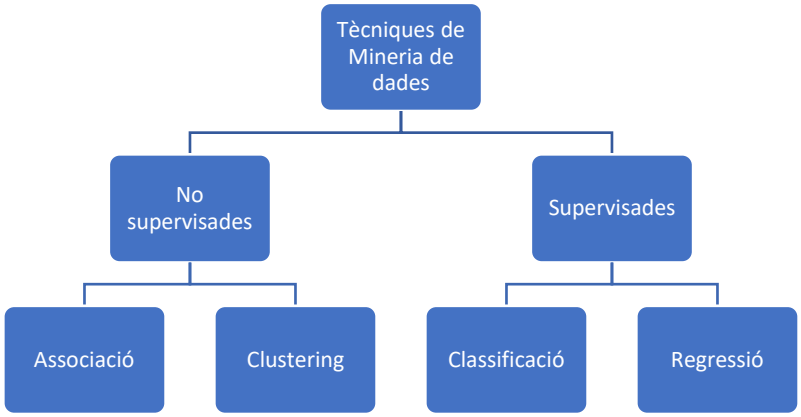


Figura 11. Tècniques de mineria de dades

Tècniques supervisades

Les tècniques de minera de dades supervisades són adequades quan es té un valor de sortida específic que es vol predir sobre les dades disponibles. S'anomena aprenentatge supervisat perquè compten amb un aprenentatge previ basat en un sistema d'etiquetes associades a les dades, amb la classe o resultat esperat.

L'objectiu de l'aprenentatge supervisat és aprendre una funció que, donada una mostra de dades i uns valors de sortida desitjats, s'aproxima millor a la relació entre l'entrada i la sortida observable de les dades. L'algorisme fa prediccions iterativament sobre el subconjunt de dades amb els valors de sortida prèviament coneguts (*Training set*) i para quan obté un cert nivell acceptable d'encerts. A continuació, s'aplica aquest model a dades de les quals es desconeix el valor objectiu (*Testing set)*.

Regressió: La variable de sortida és un valor real, com ara "*euros*" o "*quilos*". Alguns exemples d'algorismes són: regressió lineal i polinòmica i *Random Forests*

Classificació: La variable de sortida és una variable categòrica, com ara "*blau*" o "*vermell*" o "*malalt*" o "*no malalt*". Alguns exemples d'algorismes són: regressió logística, KNN (*K-nearest neighbors), Support vector machines (*SVM*)*, *Classification Trees* i *Naive-Bayes.*

[3] L'aprenentatge automàtic (*Machine Learning)* és una disciplina de l'àmbit de la Intel·ligència Artificial que a través d'algorismes, dota als ordinadors de la capacitat de identificar patrons en dades massives per fer prediccions. Aquest aprenentatge permet als computadors dur a terme tasques específiques de forma autònoma, sense necessitat de ser programats.

Tècniques no supervisades

L'aprenentatge no supervisat és la segona tècnica d'aprenentatge automàtic, on no compten amb un aprenentatge previ. L'algorisme conté variables d'entrada però no hi ha valors de sortida coneguts (etiquetes). Això vol dir que el nostre algoritme no aprèn explícitament un model, sinó que memoritza les instàncies d'entrenament que són utilitzades com a "base de coneixement" per a la fase de predicció.

L'objectiu de l'aprenentatge no supervisat és deduir les relacions ocultes i l'estructura natural present dins d'un conjunt de punts de dades. El model ha de funcionar per si mateix per descobrir informació sense utilitzar etiquetes de dades proporcionades explícitament.

Clustering: Es vol descobrir les agrupacions inherents a les dades, com ara agrupar clients mitjançant el seu comportament de la compra. Alguns exemples d'algorismes són: *K-means*, SVD i PCA.

Associació: Es vol descobrir regles que descriuen grans porcions de les dades, com ara "*les persones que compren X també tendeixen a comprar Y*". Alguns exemples d'algorismes són: *Apriori* i *FP-Growth*

L'estudi es centrarà en tècniques supervisades, concretament en mecanismes de classificació. Els classificadors, o mètodes predictius de classificació, prediuen la classe dels punts d'un conjunt de dades donats. Les classes, també anomenades etiquetes o categories, són de caràcter discret. Per exemple, la detecció de correu *spam* es pot identificar com un problema de classificació. S'utilitzaria com a conjunt d'entrenament (*Training set)* dades conegudes de correus *spam* i *no spam* per tal de comprendre la relació entre les variables d'entrada donades i la classe. Una vegada entrenat, el classificador es pot utilitzar per a detectar un correu desconegut.

L'anàlisi de dades a realitzar s'aplica a dades on la variable dependent (Y) serà una variable binària que prendrà el valor "A" si l'assignatura està aprovada o bé el valor "S" en cas que l'assignatura estigui suspesa. Les variables independents (X) seran les notes numèriques dels estudiants a la fase inicial.

El modelatge s'enfocarà en l'estudi de classificadors, la seva aplicació i posterior validació. Els algorismes seleccionats en aquest anàlisi són la regressió logística i *K-Nearest Neighbors (k-NN).* Aquests s'extrauran de la llibreria *sk-learn* en la qual es troben programats per a la seva utilització.

3.2. Regressió logística

La regressió logística és un algorisme d'aprenentatge automàtic que s'utilitza principalment per a la classificació binària. El model prediu la probabilitat d'una variable dependent Y en funció d'una o més variables independents X.

En la regressió logística la variable dependent és una variable binària que conté dades codificades com a 1 (Sí, èxit, veritable, etc.) o 0 (No, fallada, fals, etc.). En el nostre cas, codificarem la variable "S" (*Suspès*) com a classe 1 o positiva i la variable "A" (*No suspès*) com a classe 0 o negativa.

A diferència de la Regressió Lineal, en la qual la sortida és la suma ponderada de les entrades, per a la Regressió Logística s'utilitza la funció sigmoide. La Regressió Logística és una Regressió Lineal generalitzada de manera que no produïm la suma ponderada de les entrades directament, sinó que la passem a través de la funció sigmoide que pot mapar qualsevol valor real entre 0 i 1.

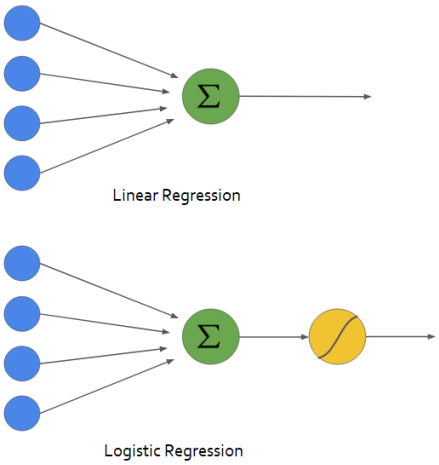


Figura 12. Regressió lineal i regressió logística

A continuació es mostra la funció d'activació coneguda com a funció o corba sigmoide (σ). És una corba en forma de S, que pot prendre qualsevol valor real X i el transforma en un valor entre 0 i 1, però mai exactament en aquests límits.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

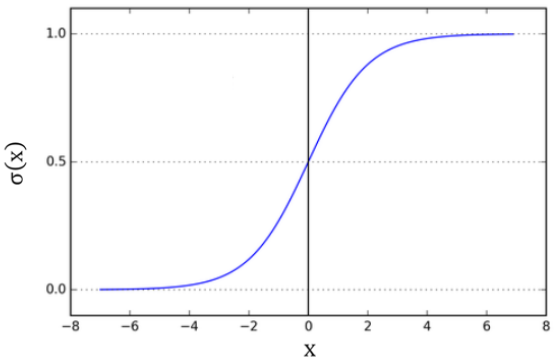


Figura 13. Funció sigmoide [4]

En la Regressió lineal s'utilitza l'equació:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

On y és la variable dependent i x_1, x_2 ,..., x_n són les variables explicatives.

[4] La funció logística s'utilitza per descriure moltes situacions del món real, com per exemple, el creixement de la població. Si s'observa el gràfic normalitzat, les etapes inicials pateixen un creixement exponencial, i al cap d'un temps, a causa de la competència per a determinats recursos (coll d'ampolla), la taxa de creixement disminueix fins arribar a l'aturada i no hi ha creixement.

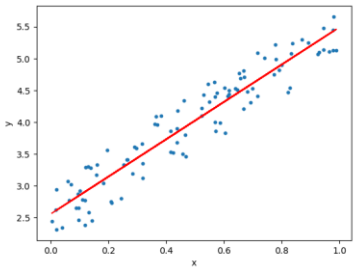


Figura 14. Regressió lineal

Si apliquem la funció sigmoide en la regressió lineal, obtenim l'equació de la Regressió Logística binària que calcula la probabilitat que una observació X pertanyi a la classe predeterminada (per exemple, la primera classe):

$$\mathrm{p(x)} = \frac{1}{1+e^{-y}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}}$$

On la funció $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$ es troba a l'eix horitzontal i p(X) a l'eix vertical.

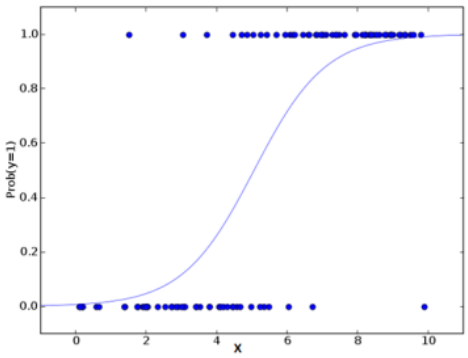


Figura 15. Regressió logística

Aquesta funció associa un valor qualsevol d' X a una probabilitat d'un esdeveniment (p), σ(X). S'observa que el valor de la funció sempre es troba entre 0 i 1. S'utilitza com a valor llindar de probabilitat el 0,5 de manera que si la probabilitat és superior o igual a 0,5 ($p \geq 0{,}5$) la classifiquem com a classe Y=1 mentre que si $p < 0{,}5$ serà de classe Y=0.

Per exemple, es vol modelar el sexe d'una població com a home o dona en funció seva alçada. La primera classe podria ser la masculina i el model de regressió logística es podria escriure com a la probabilitat de pertànyer a la classe masculina donada l'alçada d'una persona o, més formalment:

$$P\ (\text{sexe} = \text{home} \mid \text{alçada})$$

Escrit d'una altra manera, estem modelant la probabilitat que una entrada (X) pertanyi a la classe predeterminada (Y = 1), podem escriure-la formalment com:

$$P\ (X) = P\ (Y = 1 \mid X)$$

Es planteja la següent pregunta: donada una alçada (X) de 150cm, a quina classe pertany aquesta persona?

Primerament, es calcula la probabilitat de pertànyer a la classe masculina donada l'alçada de 150 cm o més formalment P (home | alçada = 150).

$$p = \frac{e^{(\beta_0+\beta_1 x)}}{1 + e^{(\beta_0+\beta_1 x)}}$$

Donats els coeficients de β_0= -100 i β_1 = 0,6:

$$p = \frac{e^{(-100+0.6\,x)}}{1 + e^{(-100+0.6\,x)}}$$

$$p = \frac{e^{(-100+0.6\cdot 150)}}{1 + e^{(-100+0.6\cdot\,150)}}$$

$$y = 0{,}0000453978687$$

S'obté una probabilitat de gairebé zero que la persona sigui un mascle.

A continuació s'ha de transformar la predicció de probabilitat en valors binaris (0 o 1). Com que es tracta d'una classificació binària, podem separar les probabilitats de la següent manera:

Classe 0 si P(Home) < 0,5

Classe 1 si P(Home) ≥ 0,5

Així doncs, podríem dir que la persona d'alçada 150cm pertany a la classe 0 o dona.

Regressió Logística vs Regressió Lineal

La regressió lineal presenta una sortida contínua mentre que la regressió logística proporciona una sortida constant. És per això que s'utilitza la Regressió Logística per a tasques de classificació. A més, a diferència de la regressió lineal, la regressió logística no fa cap supòsit de normalitat, linealitat i homogeneïtat de variància. Aquesta és una de les raons per les quals la regressió logística podria ser més potent ja que aquests supòsits són difícils de complir en el món real.

En el següent gràfic s'observa com la regressió logística, en aquest conjunt de dades, classificaria com a valors com 0 o 1 mitjançant la corba logística.

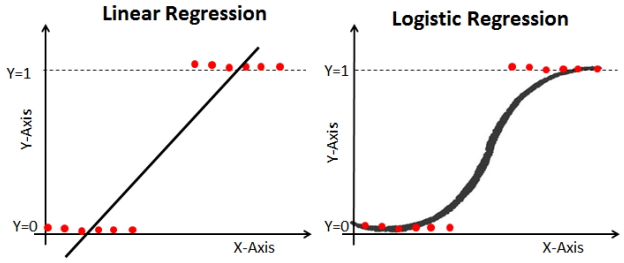


Figura 16. Regressió logística i regressió lineal

Si s'apliqués la Regressió lineal a les dades del treball, s'obtindria la predicció de la nota numèrica d'un alumne (variable continua). En la regressió logística, s'assignen els valors de *Aprovat* o *Suspès* i es prediu si l'alumne aprova o no l'examen, sense donar valors numèrics exactes.

A continuació es mostra una taula comparativa entre la Regressió Lineal i la Regressió Logística.

	Regressió Lineal	**Regressió Logística**
Definició	Prediu una variable continua dependent basada en variables independents Aproximació lineal que modela la relació entre una variable dependent i una o varies variables independents Estima la variable dependent quan la variable independent vaira	Prediu una variable categòrica dependent basada en variables independents Model estadístic que prediu la probabilitat d'un valor de sortida binari Calcula la probabilitat d'ocurrència d'un esdeveniment
Tipus de variable	Variable dependent continua	Variable dependent categòrica
Mètode d'estimació dels coeficients	Mètode dels mínims quadrats	Maximum-likelihood Estimation (MLE)
Equació	$y = \beta_0 + \beta_1 x_i$	$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_i$
Línia de regressió	Línia recta	Corba S
Resultat	Valor de sortida enter	Valor de sortida binari (0 o 1)
Aplicacions	Problemes de regressió Ex: Predicció de preus o resultats	Problemes de classificació Ex: Correu *spam* o *no spam,* Tumor *maligne* o *benigne*

Taula 8. Taula comparativa entre la regressió logística i la regressió lineal

Supòsits de la Regressió Logística:

- La regressió logística binària requereix que la variable depenent sigui binària.
- El nivell 1 de la variable depenent hauria de representar el resultat desitjat.
- Només s'ha d'incloure variables significatives.
- El model ha de tenir poca o cap multicol·linealitat. És a dir, les variables independents han de ser independents les unes amb les altres.
- Les variables independents estan linealment relacionades amb el logaritme de les probabilitats *(Log odds)*.
- La regressió logística requereix un gran nombre de mostres.

Tipus de Regressió Logística:

- Regressió Logística binària: La variable objectiu només té dos possibles resultats. Ex: *Spam* o *no spam*, malalt o no malalt.
- Regressió Logística Multinominal : La variable objectiu té tres o més categories nominals. Ex: "Malatia A", "Malaltia B" i "Malaltia C"
- Regressió logística ordinal: La variable objectiu té tres o més categories ordinals. Ex: classificació d'un producte de l'1 al 5 o classificació mitjançant "Baix", "mitjà" i "alt".

3.3. K-Nearest neighbors

L'algorisme KNN (o veïns més propers en català) és un dels algorismes de classificació més bàsics i essencials en l'aprenentatge automàtic. Pertany al domini d'aprenentatge supervisat i troba una intensa aplicació en el reconeixement de patrons, la mineria de dades i la detecció d'intrusions. Es pot utilitzar tant per a problemes de predicció de classificació com de regressió. Tot i així, s'utilitza més àmpliament en problemes de classificació de la indústria.

L'algorisme assumeix que els elements similars es troben a poca distància.

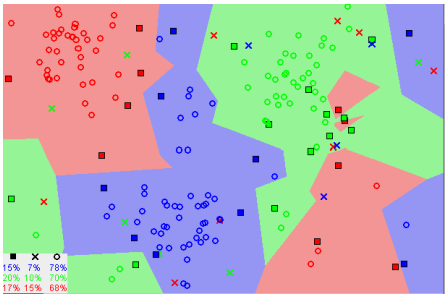


Figura 17. Exemple de distribució de classes segons la proximitat

A partir de la imatge superior s'observa com la majoria de les vegades, els punts de dades similars es troben a prop els uns als altres. L'algorisme depèn que aquesta suposició sigui prou certa perquè sigui útil.

KNN és un mètode que emmagatzema tots els casos disponibles i classifica els nous basant-se en una mesura de semblança, la proximitat. Assigna la nova observació a la classe més comuna d'entre els seus veïns més propers k, mesurats per la funció distància.

KNN presenta dues propietats importants:

- **Algoritme d'aprenentatge mandrós**: KNN és un algorisme d'aprenentatge mandrós perquè no té una fase d'entrenament especialitzada i utilitza totes les dades per a l'entrenament mentre es classifica. És a dir, no aprèn una funció discriminatòria del conjunt de dades d'entrenament, simplement les memoritza.

  Cal destacar que la fase mínima d'entrenament té un cost de memòria elevat ja que s'ha d'emmagatzemar un conjunt de dades enorme, així com un cost computacional durant el temps de prova, doncs per classificar una nova observació necessita utilitzar tot el *Dataset*. Això no és desitjable perquè a la pràctica es volen respostes ràpides. Per aquestes raons, KNN funciona millor en *Datasets* petits i amb dimensions baixes.

- **Algorisme d'aprenentatge no paramètric:** no assumeix res sobre les dades subjacents. Està àmpliament disponible en escenaris de la vida real, ja que no fa cap suposició subjacent sobre la distribució de dades

Es pren un cas senzill per entendre el funcionament d'aquest algorisme. Es pren un conjunt de dades que es pot representar de la següent manera:

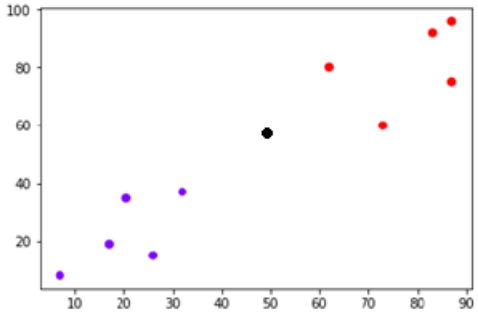


Figura 18. Conjunt de dades d'entrenament amb un nou punt de dades

Ara, donat un altre conjunt de punts de dades (també anomenades dades de prova o *Testing set*), s'assignen aquests punts a una classe analitzant el conjunt d'entrenament. Es classifica el nou punt de dades de color negre (50,60) en classe blava o vermella observant a quin grup pertanyen els seus veïns més propers. Suposem k=3 que és el nombre de veïns més propers. Es traça un cercle amb el punt negre com a centre tan gran com per incloure només tres punts de dades d'entrenament del pla.

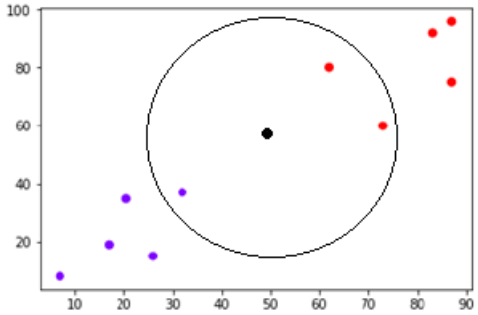


Figura 19. Classificació del nou punt per k=3

A partir de la figura anterior s'observen els tres veïns més propers del punt de dades negre. Entre aquests tres, dos d'ells pertanyen a la classe vermella, de manera que el punt negre s'assignarà a la classe majoritària, la vermella.

Ara bé, si triem k=5 el punt pertany a la classe blava. Més endavant s'explica com triar el valor d'aquest paràmetre k.

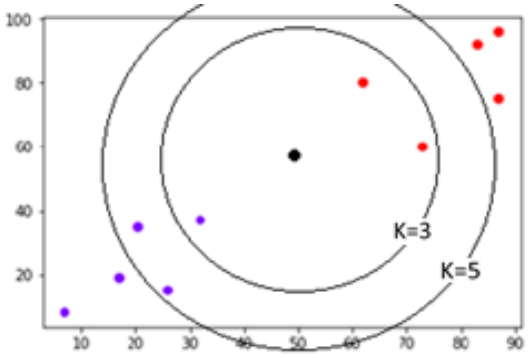


Figura 20. Classificació del nou punt per k=3 i k=5

Algoritme

L'algoritme KNN utilitza la "semblança de característiques" per predir els valors dels nous dipòsits de dades, cosa que significa que al nou punt de dades se li assignarà un valor basat en la proximitat amb els punts del conjunt d'entrenament. Podem entendre el seu funcionament amb l'ajuda de l'exemple següent:

Es consideren les següents dades sobre la detecció de fraus en les targetes de crèdit. L'edat i el préstec són dues variables numèriques (predictors) i cada punt es classifica com a frau (S, *Default*) o No frau (N, *Non-Default*).

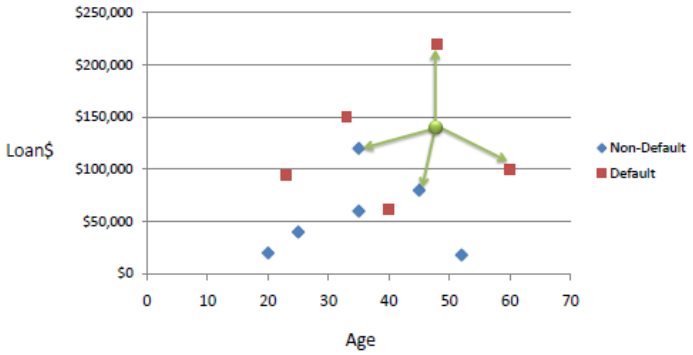


Figura 21. Distribució de dades segons les variables edat i préstec.

1. Per implementar qualsevol algorisme, es necessita un conjunt de dades. Així, durant el primer pas de KNN, s'ha de carregar el *Testing set* i *Training set.*

2. A continuació, s'ha de triar el valor de k és a dir, els punts de dades més propers. K pot ser qualsevol nombre enter, es considera k=3.

3. Sigui **m** el nombre de mostres de dades d'entrenament (*Training set)*. Sigui **p** un punt desconegut de les dades de prova (Edat = 48 i Préstec = 142.000 $):

    a. Calcula la distància[5] entre el punt a classificar i cada fila de dades d'entrenament. S'utilitzarà la fórmula més comuna per calcular-la, la Euclidiana:

$$Distància\ Euclidiana = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

On $(x_A, y_A) = (48\,, 142.000)$

Edat (x)	Préstec (y)	Classe	Distància Euclidiana
25	40.000 $	N	102.000
35	60.000 $	N	82.000
45	80.000 $	N	62.000
20	20.000 $	N	122.000
35	120.000 $	N	22.000
52	18.000 $	N	124.000
23	95.000 $	S	47.000

[5] També es poden utilitzar els mètodes Manhattan o Minkowski per a variables continues. En el cas de variables categòriques s'utilitza la distància de Hamming.

40	62.000 $	S	80.000
60	100.000 $	S	42.000
48	220.000 $	S	78.000
33	150.000 $	S	8.000
48	142.000 $	?	

Taula 9.Càlcul de la distància Euclidiana entre el punt de dades i el conjunt d'entrenament

Si es calcula la distància entre p i l'últim cas del conjunt d'entrenament:

$$Distància = \sqrt{(48-33)^2 + (142000-150000)^2} = 8000{,}01$$

b. En funció del valor de la distància, s'ordenen en ordre ascendent i es tria les files K principals de la matriu ordenada.

Si k=3:

Edat (x)	**Préstec (y)**	**Classe**	**Distància Euclidiana**
33	150.000 $	S	8.000
35	120.000 $	N	22.000
60	100.000 $	S	42.000
23	95.000 $	S	47.000
	...	...	
48	142.000 $	S	

Taula 10. Determinació de la classe del nou punt de dades

c. Ara s'assignarà una classe al punt de prova en funció de la classe més freqüent d'aquestes files.

Amb k=3, hi ha dues classes = S i una classe =N dels tres veïns més propers. La predicció del nou punt de dades és classe=S.

4. Repetir el procediment per a cada punt de les dades de prova (*Testing set).*

Un dels principals inconvenients en el càlcul de mesures de distància directament del conjunt d'entrenament és en el cas en què les variables tinguin diferents escales de mesura o hi hagi una barreja de variables numèriques i categòriques. En el exemple anterior, la variable dels préstecs en dòlars presenta una influència molt més gran en la distància calculada en front a l'altre variable, l'edat. Una solució és normalitzar el conjunt d'entrenament de manera que les variables prenguin el mateix rang de valors.

Valor del paràmetre k

Abans d'escollir k, cal definir dos conceptes importants en l'aprenentatge de màquines:

- **El biaix** és la diferència entre el valor correcte i la nostra predicció (o mitjana) esperada[6]. És un error derivat de supòsits erronis de l'algoritme d'aprenentatge. Un biaix elevat pot fer que un algorisme perdi les relacions rellevants entre les funcions i les sortides de destinació.
  El model amb un biaix elevat presta molt poca atenció a les *Training data* i sobre simplifica el model.

$$Biaix\ (x) = E[\hat{f}(x)] - f(x)$$

- **La variància**, en teoria de probabilitat i estadística, es una mesura de dispersió definida com l'esperança del quadrat de la desviació d'una variable aleatòria respecte la seva mitjana. En altres paraules, la variància és la quantitat que varien les prediccions per a un punt determinat entre diferents realitzacions del model. És un error de sensibilitat a petites fluctuacions en el conjunt de dades d'entrenament.
  Una alta variància presta molta atenció al *Training data* i no generalitza les dades que no han vist abans.

$$Var(x) = E[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2]$$

Es poden traçar quatre casos diferents que representen combinacions de biaix i variància elevada i baixa. El centre de l'objectiu (part vermella) són els valors correctes de les dades. A mesura que s'allunya d'aquesta regió, l'error es fa més gran. En aquest cas, s'obté un biaix més alt. Ara bé, si obtenim diverses prediccions basades en la variabilitat del conjunt de dades d'entrenament s'obté una variància més alta.

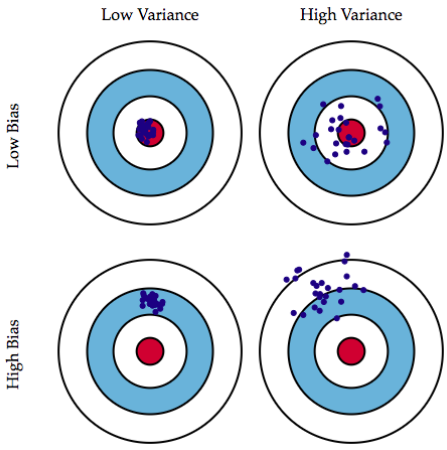


Figura 22. Il·lustració gràfica del biaix i la variància.

[6] Per cada model s'obté un valor de predicció esperat. Ara bé, si es repeteix el procés de creació del model més d'una vegada (recopilació de dades i execució d'una nova anàlisi creant un nou model), els models resultats tindran diverses prediccions a causa de l'aleatorietat dels conjunts de dades subjacents. El biaix mesura la distancia de la predicció d'aquests models en relació al valor correcte.

Fer front al problema del biaix i la variància consisteix a tractar amb els termes *Overfitting* i l'*Underfitting:*

- ***Underfitting*:** el model no s'ajusta, és a dir, no prediu correctament les dades d'entrenament donant lloc a una baixa generalització. Normalment succeeix quan es tenen poques dades per construir un model precís o quan s'intenta construir un model lineal amb dades no lineals. El model serà excessivament senzill amb moltes prediccions errònies.

  Implica biaix elevat i variància baixa.

  Overfitting: el model prediu massa bé les dades d'entrenament però és molt dolent en predir noves dades. Normalment succeeix quan s'entrena el model amb moltes dades i aprèn els detalls i el soroll d'aquestes. El problema és que aquests conceptes no s'apliquen a les dades noves i afecten negativament la capacitat de generalització dels models. Sovint és causat per un model excessivament complex.

  Implica biaix baix però variància elevada.

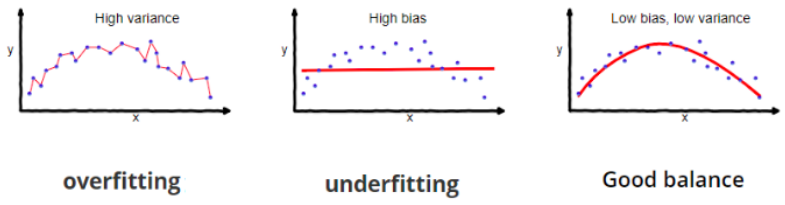


Figura 23. Biaix-Variància – Underfitting-Overfitting

El model de l'esquerra és el més complex, capta tots els punts de dades, però presenta una gran variància. El model del mig és més simple i presenta un biaix elevat. El model adequat té baix biaix i poca variància, que és el que es desitja.

Aplicació dels conceptes

Mitjançant el paràmetre K es poden demostrar aquestes idees de *Underfitting*, *Overfitting* i generalització. La següent taula resumeix algunes característiques dels models en relació amb els conceptes apresos:

	Model				
	Bias	Variance	Complexity	Flexibility	Generalizability
Underfitting: you have an overly simple model	High	Low	Low	Low	High
Overfitting: your model is modelling the noise	Low	High	High	High	Low

Figura 24. Característiques dels models en funció de Over- o Under-fitting.

Per a posar en pràctica aquests conceptes, s'exposa el següent problema de classificació binària on cada punt del conjunt de dades es classifica com a blau o vermell. Donat el següent conjunt de dades:

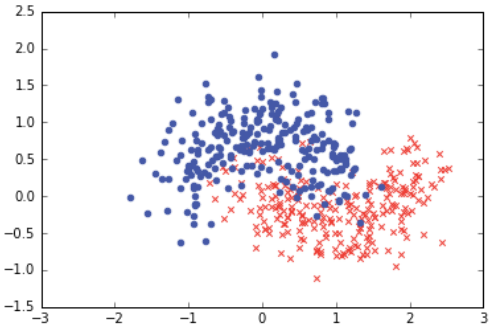


Figura 25. Distribució del conjunt de dades binari

Es divideix el conjunt de dades en un conjunt d'entrenament i un conjunt de proves.

- El ***Training set*** s'utilitzarà per desenvolupar i entrenar el model. Pren com a entrada les dades d'entrenament (X) i el valor objectiu corresponent (y) i produeix un model après (*h*).

- El ***Test set*** quedarà completament sol fins al final, moment en el qual es podran executar els models acabats. Pren com a entrada observacions noves i utilitza la funció *h* per produir les seves corresponents respostes.

Fase d'entrenament

Tal i com s'ha mencionat anteriorment, el KNN és un algorisme mandrós. La fase d'entrenament es redueix a memoritzar les dades d'entrenament que s'utilitzen posteriorment com a "coneixement" per a la fase de predicció. Concretament, això vol dir que només quan es faci una consulta a la nostra base de dades (és a dir, quan es demani la predicció d'un nou punt de dades), l'algorisme utilitzarà les instàncies d'entrenament per deixar anar una resposta.

Per a una comprensió visual, es pot pensar en l'entrenament de KNN com un procés per acolorir regions i traçar límits al voltant de les seves dades:

En dues dimensions, existeix una línia de punts[7] exactament a la meitat del camí entre dos punts de dades d'entrenament. Qualsevol punt d'aquesta línia és equidistant als dos exemples d'entrenament diferents. Si es tractés de dades tridimensionals, hi hauria un pla entre els dos punts i, per a dades de dimensions més altes, un hiperplà. Però l'important d'aquesta línia/hiperplà és que tot el que queda a l'esquerra d'aquesta estarà més a prop del punt blau (i es classificarà blau), mentre que tot el de la dreta estarà més a prop del punt verd.

[7] Mediatriu del segment d'unió entre els dos punts.

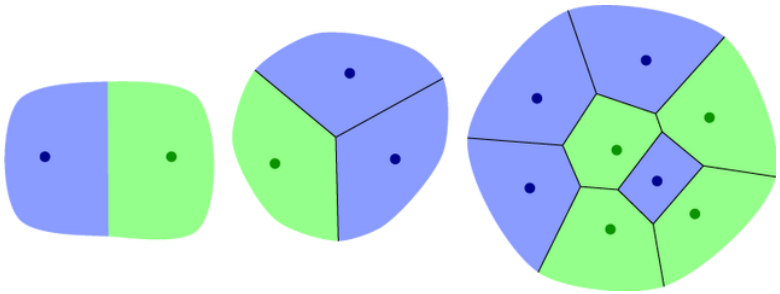


Figura 26. Exemple de cel·les de Voronoi

Si tenim tres o més punts de dades s'observa que aquest mateix tipus de comportament es repeteix, amb línies entre parells de punts, que conflueixen en un punt que es troba a la mateixa distància dels tres punts de dades.

Les línies al voltant de cada punt de dades d'entrenament conformen un polígon anomenat cel·la de Voronoi, format per el conjunt de punts més propers a aquest punt de dades que a qualsevol altre punt. Cada cel·la de Voronoi obté una classificació (en aquest cas, blau o verd), definida per la classe del punt d'entrenament que la defineix. Qualsevol punt de dades nou que es trobi dins la cel·la de Voronoi se li assignarà aquella classe. Si es pren una col·lecció d'aquestes cel·les, s'anomena tessel·lació o diagrama de Voronoi.

El límit de decisió és la línia que separa les cel·les que s'associen a una classe i les cel·les associades a una altra classe, és a dir, el llindar el qual l'algorisme decideix si un punt de dades pertany a la classe blava o verda.

La següent figura mostra cel·les de Voronoi per cada punt del conjunt d'entrenament, és a dir, quan k=1:

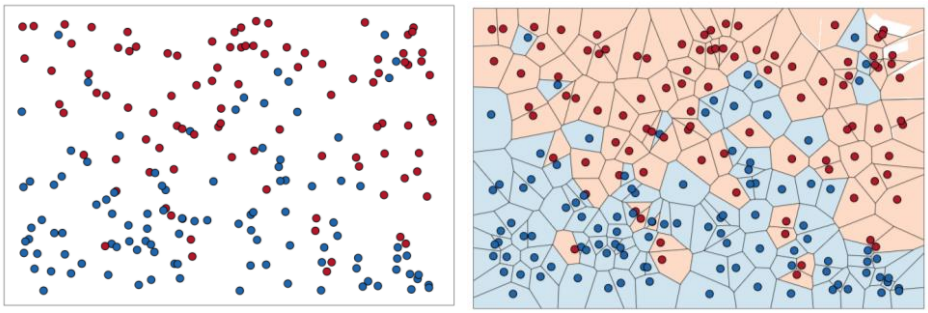


Figura 27. Exemple de diagrama de Voronoi per k=1

Així doncs, s'ha traduït l'algorisme de KNN en una imatge geomètrica en que cadascun dels punts de dades d'entrenament es troben envoltats d'una cel·la de Voronoi formada per tots els punts de l'espai els quals el seu veí més proper és aquest punt de dades.

Fins ara, s'ha utilitzat k=1. Tanmateix, k és un valor ajustable que pot prendre valors des de 1 fins al nombre de punts de dades del conjunt d'entrenament (n). Cal afegir que el valor de k no ha de ser un múltiple del nombre de classes. Per a un problema de classificació binària, es selecciona un valor imparell de k per evitar confusions entre dues classes de dades.

Per a una k diferent de 1, les arestes entre les cel·les ja no es defineixen només pels dos punts més propers. La següent imatge, mostra una aproximació de la distribució resultant quan k=3.

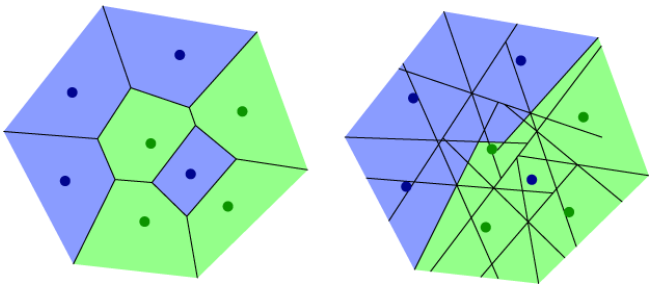


Figura 28. Cel·les de Voronoi per k=1 (esquerra) i k=3 (dreta)

A primera vista, s'observen dues diferències importants entre els diferents valors de k. La primera és que per a k diferent a 1 es troben més línies implicades a la imatge i l'estructura general és més senzilla, ja no hi ha un forat al mig de la regió verda i s'observa com el límit de decisió entre la classe blava i verda es suavitza. En segon lloc, per a k=3 s'observa que hi ha un punt blau totalment dins una regió verda.

Una vegada introduït com l'algorisme classifica els nous punts mitjançant regions, es posarà en practica mitjançant l'exemple anterior. A partir del *Training set* del conjunt de dades s'ajustarà el valor k per a dos casos extrems.

Training set

Quan k= 1, per a cada punt de dades, x, en el conjunt d'entrenament, es vol trobar un altre punt, x', que tingui la menor distància de x. La distància més curta possible sempre és 0, de manera que el "veí més proper" és en realitat el punt de dades original, x = x '. Per aquest motiu, l'error d'entrenament serà zero quan k= 1, independentment del conjunt de dades.

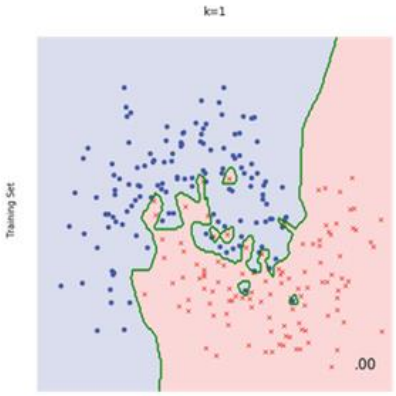


Figura 29. Distribució de dades del Training set per k=1

S'observa com no hi ha punts vermells a les regions blaves i viceversa, s'observen petites "illes". Tècnicament està generant prediccions perfectament correctes sobre el conjunt d'entrenament però s'observa que aquest ajustament és massa sensible als punts de dades individuals.

És un model molt flexible i amb molta complexitat: s'ajusta molt a la forma precisa del conjunt de dades. Sembla que el model s'adapta massivament al soroll.

El límit de de decisió de les dades de formació (representat per la línia verda) presenta moltes irregularitats. Per a k petites, les irregularitats i les illes són signes de variància.

També té un biaix baix: si no hi ha res més, el límit de decisió s'ajusta a les tendències que s'observen a les dades.

Quan k=99, s'acoloreixen les regions al voltant d'un punt d'entrenament en funció de la categoria d'aquest i la dels seus 98 veïns més propers. Si la majoria de veïns són blaus, però el punt original és de color vermell, la regió pren el color de la majoria blava. És per això que es poden tenir punts de dades vermells en una regió blava i viceversa. Per tant, l'error d'entrenament serà diferent a zero (indicat a la part de baix a la dreta de la imatge).

El model està força restringit, ja que ha de tenir en compte una gran quantitat d'informació a l'hora de classificar les instàncies. En altres paraules, un nombre elevat de k dona lloc a un comportament relativament "rígid".

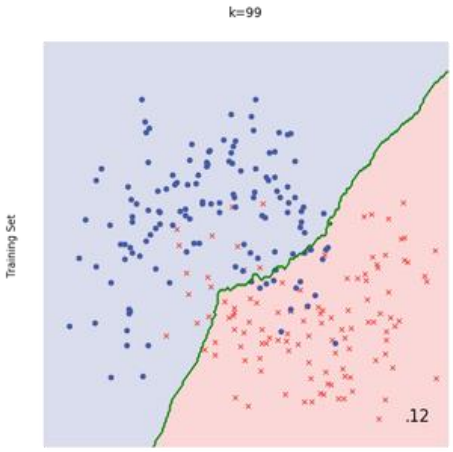


Figura 30. Distribució de dades del Training set per k=99

S'observa que el model presenta un límit de decisió més suau i les illes desapareixen.

El model presenta poca flexibilitat i baixa complexitat: la distinció entre les dues categories es difumina i la línia de predicció de límits no es correspon en absolut.

Té un biaix relativament elevat, ja que es pot dir que no està modelant les dades tan bé com podria, modela les dades de manera massa simple, i això està molt esbiaixat de la realitat. És un model estable que no varia gaire, però té una baixa variància.

Testing set

Com de bé es generalitzen aquests models, és a dir, com es comporten amb les dades noves?

Fins ara només s'han vist les dades d'enterament, però quantificar l'error d'entrenament no és gaire útil. No interessa mesurar la capacitat dels models en recapitular el que acaben de veure a l'entrenament, sinó com els models funcionen amb les dades del conjunt de prova, obtenint una estimació del rendiment del model.

A continuació es mostren els límits de decisió apresos mitjançant el conjunt d'entrenament aplicats al conjunt de proves.

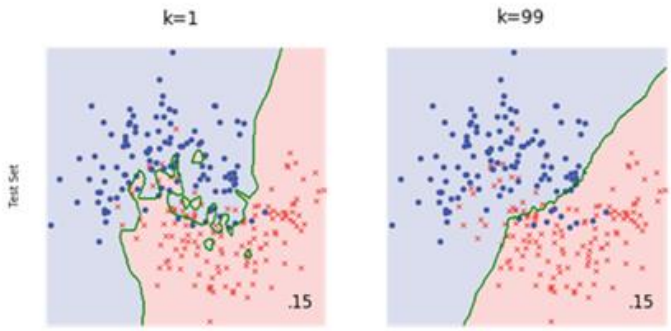


Figura 31. Distribució de dades del Testing set per k=1 i k=99

Els dos models s'equivoquen per raons completament diferents:

El model amb k=1 està cometent errors per haver-se adaptat massa al soroll, *Overfitting.* Cal recordar que l'*Overfitting* presenta un bon rendiment de les dades d'entrenament però una mala generalització de les dades de prova, que és exactament el que es pot observar. Per altra banda, el model amb k=99 no s'adequa a la forma creixent de les dades, hi ha *Underfitting*

Ara bé, si es prova un valor de k entre 1 i 99 com per exemple k=50:

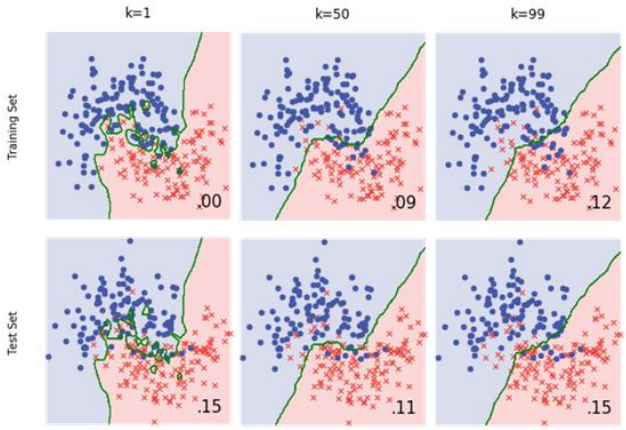


Figura 32. Comparació del valor de k sobre el Testing set i Training set

S'observa que l'adequació del model és similar a la tendència real del conjunt de dades. Aquesta millora es veu reflectida en un error del conjunt de proves inferior (de 0,11 en front a 0,15).

La imatge anterior mostra com el límit es suavitza a mesura que creix k. Amb l'augment de k de 1 fins a al nombre total de punts de les dades d'entrenament, on finalment es torna tot blau o vermell en funció de la majoria total.

Resumint, quan s'entrenen algorismes d'aprenentatge automàtic, el que interessa és el rendiment del model en un conjunt de dades independent. Essencialment, interessa construir models generalitzables. És a dir, no s'ha d'intentar modelar les tendències del conjunt de dades, sinó el procés del món real que ens ha portat a observar les dades, ja que el conjunt de dades específic amb el que es treballa no és més que una petita mostra d'instàncies de la realitat, amb el seu propi soroll i peculiaritats.

Tipus d'errors

La figura següent (*Fig. 32*) mostra el comportament típic de l'error de prova i d'entrenament front a la complexitat del model. La complexitat d'un model ve determinada pel nombre de paràmetres efectius. En l'algorisme de KNN la complexitat està controlada per k, doncs el nombre de paràmetres efectius o graus de llibertat (λ) es defineix com n/k on n és la mida del conjunt d'entrenament. Així doncs, la complexitat disminueix amb l'augment de k. Això es deu perquè l'enfocament KNN divideix la regió formada pels punts d'entrenament en n/k parts (o cel·les) on cada una està governada pel classificador de la majoria dels vots.

k=1 es el model més complex doncs presenta el major nombre de paràmetres efectius (n) i els límits de decisió més revoltats. Si es té un conjunt de dades d'entrenament de mida n=50, l'elecció de k=25 proporciona un nombre de paràmetres efectius=2.

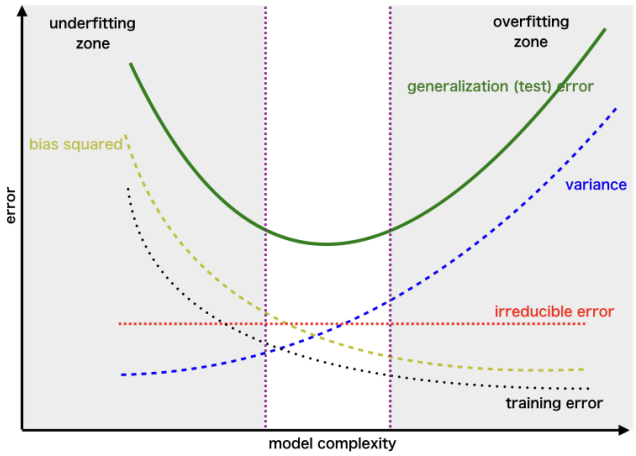


Figura 33. Gràfic de la complexitat del model vs l'error

L'error de predicció esperat per a nou punt de dades x_0, també conegut com a error de prova o de generalització, presenta la forma d'una U invertida. Es pot descompondre com :

$$Err(x_0) = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_i) \right)^2 + \frac{\sigma_\epsilon^2}{k} + \sigma_\epsilon^2$$

$$Err(x_0) = Biaix^2 + Variància + Error\ irreductible$$

On x_i és la seqüència dels veïns més propers a x_0. **[2]**

El terme de la variància correspon a la seva mitjana i està en funció de l'error irreductible σ_ϵ^2 i del valor k, de manera disminueix quan k augmenta. El terme del biaix augmenta amb la k. Per a k's petites, els pocs veïns propers prendran valors $f(x_i)$ similars a $f(x_0)$, de manera que la seva mitjana s'acostarà a $f(x_0)$. Quan k augmenta, els veïns estan més lluny i pot passar qualsevol cosa. Així doncs, si k varia, es produeix una compensació entre el biaix i la variància. A mesura que augmenta la complexitat del model, la variància acostuma a augmentar i el biaix al quadrat tendeix a disminuir. El comportament contrari es produeix a quan disminueix la complexitat del model.

L'error d'entrenament tendeix a disminuir sempre que augmentem la complexitat del model, és a dir, sempre que "encaixem les dades amb més intensitat". Tanmateix, si el model s'adapta massa a les dades de formació i no es generalitzarà bé, hi haurà *Overfitting* i l'error de prova

augmentarà. En aquest cas, les prediccions presentaran una gran variància tal i com es mostra al segon terme de l'expressió de l'error de prova. Per altra banda, si el model és molt simple (ks elevades) es produirà *Underfitting* i tindrà un gran biaix, tampoc es podrà generalitzar correctament.

En definitiva, trobar el balanç entre *l'Overfitting* i *l'Underfitting* consisteix en la compensació del biaix i la variància.

A partir de les corbes anteriors (*Fig. 33)* s'observa que el nivell de complexitat o la k que proporciona el millor error d'entrenament no és la mateixa que proporciona el millor error de prova.

Tanmateix, el que interessa és la k que correspon al *Test error* més baix ja que el que preocupa és el rendiment del model sobre les noves dades. Es podria pensar en realitzar repetides mesures de l'error de prova per a diferents valors de k per tal de trobar aquest valor mínim. No obstant, això podria produir *Overfitting* en el conjunt de prova ja que, involuntàriament, el que es fa és utilitzar el Test set com a *Training set*. La informació del conjunt de proves es pot filtrar inadvertidament a la fase de creació del model de manera que el aquest és incapaç de generalitzar noves observacions. Per tant, no s'ha de tocar el *Test set* fins al final del modelatge.

Així doncs, no es té accés a les dades del *Test* en la fase de modelatge ni, per tant, al valor òptim de k. Tampoc es pot escollir el error mínim d'entrenament, doncs estaríem utilitzant un valor sub-òptim de k, en concret k=1.

S'utilitzarà un mètode per ajustar el paràmetre k sense necessitat d'accedir al *Test set*. La idea és utilitzar només el *Training set* per imitar el procés d'ajustar un model a les dades d'entrenament i després aplicar-les a les dades de prova.

Es prendrà un subconjunt del conjunt d'entrenament, anomenat **conjunt de validació**, el qual s'utilitzarà per seleccionar el nivell adequat de flexibilitat[8] o complexitat del nostre algorisme. Existeixen diferents enfocaments de validació dels quals explorarem un dels més utilitzats: la validació creuada o *K-Cross Validation*.

K-Cross Validation

La validació creuada consisteix en dividir aleatòriament el conjunt d'entrenament en K grups o seccions iguals (la K[9] no està relacionada amb la k de KNN). El primer grup K_{cv} es tracta com un conjunt de validació i s'entrena l'algorisme a les altres K_{cv}-1 seccions restants. A continuació, s'avalua la el rendiment de la secció K_{cv} retinguda, com per exemple l'error. Aquest procediment es repeteix K_{cv} vegades; cada vegada es pren una secció diferent com a conjunt de validació. Finalment, s'obtenen K_{cv} estimacions diferents de l'error de validació i es calcula la mitjana d'aquests valors per a obtenir una mesura general del rendiment del model.

A continuació es mostra una imatge de validació creuada per una K_{cv}=5:

[8] 1/k.

[9] Denotarem K_{cv} (K corresponent a *Cross Validation*) per no confondre-ho amb la k de KNN.

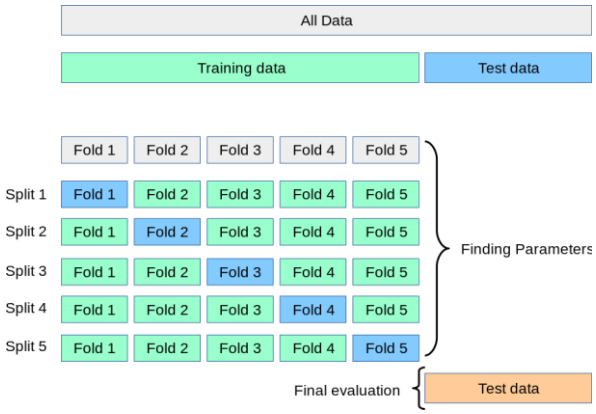


Figura 34. Procés de K-Cross Validation per a l'optimització de paràmetres

El procediment de validació creuada és realitza per a diferents valors de k, de manera que per cada k s'obté un error de validació mitjà.

Es provaran diferents valors de k a l'hora de crear models amb les dades d'entrenament i es seleccionarà el valor de k amb el millor error de validació creuada. A continuació es mostra un exemple de la corba d'error de validació creuada aproximada:

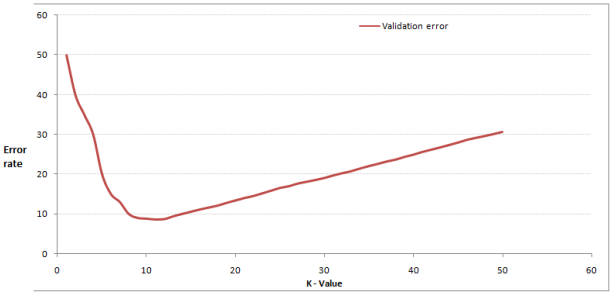


Figura 35. Corba d'error de validació

La corba d'error de validació creuada en funció k s'assembla molt més a la corba d'error de prova que a la corba d'error de formació; té una forma d'U invertida. Així doncs, en aquest cas, es prendria com a k òptima la que correspon a un error de validació creuada més baix, k=10.

Es sap que el valor k que es vol trobar és el que sigui més adequat per a les dades de prova. Com que no es té accés a aquesta k, s'espera que el conjunt de validació o la validació creuada s'acosti el màxim possible a aquest valor òptim k.

Es pot comparar el valor de k òptim per a cada procediment:

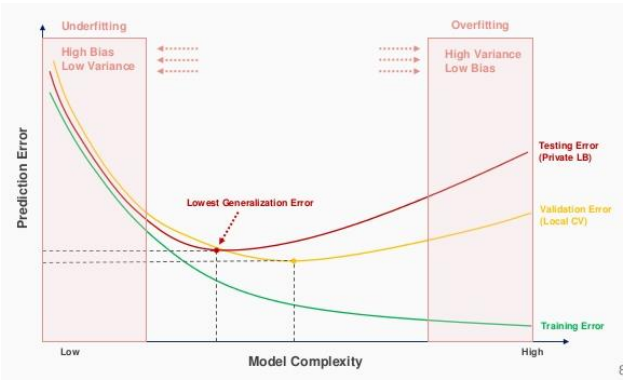


Figura 36. Corbes d'error de validació, error de prova i error d'entrenament.

S'observa que tot i que la validació creuada i l'avaluació del conjunt de prova condueixen a una òptima diferent, ambdós valors són força raonables i es troben dins del rang desitjat.

Tal i com s'ha mencionat anteriorment, no s'utilitza el conjunt de proves ni el conjunt d'entrenament per a trobar el valor òptim de k. Aquests càlculs només serveixen com a il·lustració per veure com es veurien les seves respectives corbes *Testing error i Training error.* A la pràctica, només la validació proporciona un valor de k aproximat segur i no es mostrarien les altres corbes.

4. VALIDACIÓ

4.1. Mètodes de validació

Un cop construït el model, es necessita avaluar la seva precisió i la qualitat de les dades. Per fer-ho s'utilitzaran tècniques de validació creuada.

La base de les tècniques de validació creuada és dividir el conjunt de dades a l'hora d'entrenar el model. Algunes de les dades s'eliminen abans de començar l'entrenament. Una vegada entrenat, es recuperen les dades que es van treure per avaluar el rendiment del model davant de dades que no ha vist mai. Definim aquests subconjunts com:

- ***Training set*** *(o Conjunt d'entrenament)*: Subconjunt de dades que s'utilitza per entrenar el model a partir del qual l'algoritme "aprèn" les relacions entre les funcions i la variable objectiu.

- ***Testing set*** *(o Conjunt de proves)*: Subconjunt de dades que proporciona una estimació final del rendiment del model després de ser entrenat.

Les dues estratègies més conegudes per la validació creuada són: *Holdout method* i *K-cross Validation.*

Holdout

Holdou method (o mètode de retenció en català) és el tipus més senzill de validació creuada. El conjunt de dades es separa en dos subconjunts anomenats: *Training set* i *Testing set / Holdout set.* L'estimador de funcions s'ajusta a una funció utilitzant només el *Training set* i se li demana que predigui els valors de sortida de les dades del *Testing set*, conjunt que no ha vist mai abans.

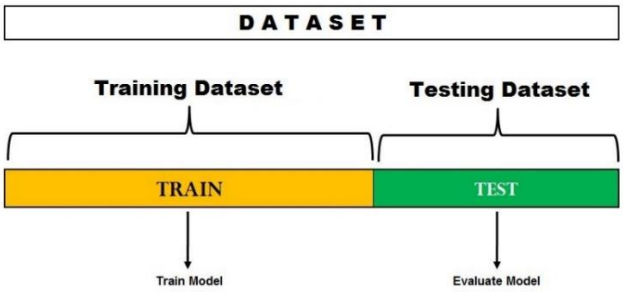


Figura 37. Mètode Holdout

Avantatges:

- Dades totalment independents. Només s'ha d'executar una vegada de manera que es redueixen els costos computacionals.

Desavantatges:

- L'avaluació pot dependre molt de quins punts de dades acaben al *Testing set* o al *Training set.* Segons com es faci aquesta divisió l'avaluació podria resultar diferent.
- L'avaluació del rendiment està sotmesa a una major variància a causa de la reduïda mida de les dades.

K-Cross Validation

K-cross validation és una millora del mètode anterior *Holdout.* El conjunt de dades es divideix en K subconjunts. Un dels subconjunts K s'utilitza com a *Testing set* mentre que els altres subconjunts K-1 s'uneixen per a formar el *Training set.* El procés de validació creuada es repeteix durant K iteracions de manera que tots els subconjunts són considerats una vegada com a conjunt de proves. Finalment, es realitza la mitjana aritmètica dels resultats de cada iteració per obtenir un únic resultat.

Si K=5:

Figura 38. Mètode de validació creuada

Avantatges:

- La divisió de les dades presenta menys importància. Cada punt de dades es troba en el *Testing set* una vegada i forma part del *Training set* K-1 vegades.
- Presenta menys variància ja que utilitza tot el conjunt com a entrenament. La variància de l'estimació resultant es redueix a mesura que s'augmenta la K.

Desavantatges:

- Costos computacionals més elevats. El model s'ha d'entrenar K vegades durant la fase de validació.

S'ha decidit utilitzar el mètode de validació *K-Cross Validation* per a la validació tant per la Regressió Logística com per KNN.

4.2. Mètriques de rendiment

Per tal d'avaluar les prediccions dels models, s'utilitzaran algunes mètriques bàsiques de rendiment.

- **Matriu de confusió**

Una matriu de confusió es una taula que s'utilitza per descriure el rendiment d'un model de classificació en un conjunt de dades de prova de les quals es coneixen els valors reals. S'anomena matriu de confusió perquè fa que sigui fàcil detectar on el model està confonent les classes.

Es recorda que les classes de la variable objectiu d'aquest treball són:

- ***Classe 1 (Positiu):*** Suspès
- ***Classe 0 (Negatiu):*** No Suspès

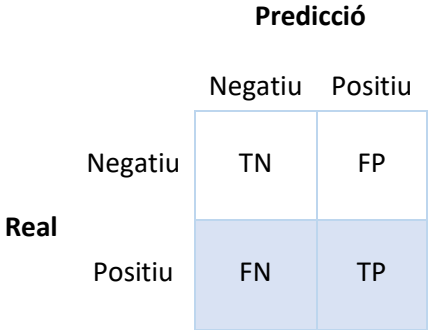


Figura 39. Matriu de confusió

Els elements de la matriu són:

- **True Positive (TP):** Valor real positiu i valor de predicció positiu. Suspès etiquetat com a suspès.
- **True Negative (TN):** Valor real negatiu i valor de predicció negatiu. Aprovat etiquetat com a aprovat.
- **False Positive (FP):** Valor real negatiu però valor de predicció positiu. Aprovat etiquetat com a suspès.
- **False Negative (FN):** Valor real positiu però valor de predicció negatiu. Suspès etiquetat com a aprovat.

A partir dels valors de la matriu de confusió, es defineixen les següents mètriques per a la classe positiva:

1. **Accuracy (Precisió)**[10]

És la mesura de tots els casos correctament identificats. S'utilitza quan la distribució de classes està equilibrada, és a dir, les dues classes presenten aproximadament el mateix nombre d'instàncies. També assumeix que els errors per falsos positius i falsos negatius tenen el mateix cost i dona més impotència als verdaders positius i verdaders negatius.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total}$$

[10] La taxa d'error és el complementari de la precisió (*1-Accuracy),* doncs es calcula com: FP+FN/Total.

2. Precision (Exactitud)

L'exactitud mesura de tots els valors positius predits, quants d'ells són realment positius. És una bona mesura quan hi ha un gran nombre de falsos positius. Pot prendre valors entre 0 i 1, essent 1 el millor valor.

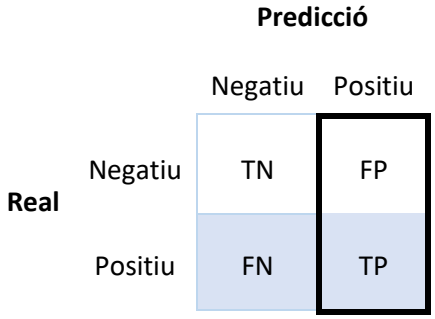


Figura 40. Matriu de confusió

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{True\ Positive}{Total\ Predicted\ Positive}$$

3. Recall (Sensibilitat)

El paràmetre *Recall* és la mesura dels casos positius identificats correctament de tots els casos positius. És una bona mesura quan hi ha un gran nombre de falsos negatius. Pot prendre valors entre 0 i 1, essent 1 el millor valor.

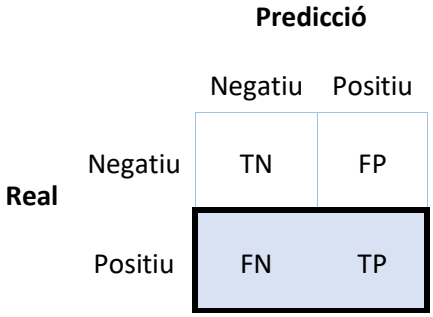


Figura 41. Matriu de confusió

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{True\ Positive}{Total\ Actual\ Positive}$$

4. F1 Score (Valor F)

F1 es podria considerar com una mitjana ponderada dels paràmetres *Precision* i *Recall.* És una bona mesura si es necessita establir un equilibri entre aquests paràmetres i existeix una distribució de classes desequilibrada.

F1 s'utilitza quan els falsos negatius i falsos positius presenten un cost molt elevat. Pot prendre valors entre 0 i 1, essent 1 el millor valor.

$$F1\ Score = 2\ x\ \frac{Precision *\ Recall}{Precision + Recall}$$

Exemple

Es considera un *Dataset* de 100 exemples, 98 dels quals pertanyen a la classe majoritària negativa (aprovats) i 2 a la classe minoritària positiva (suspesos). Així doncs, es tracta d'un conjunt de dades desequilibrat amb una rati entre classes de 1:49.

		Predicció		
		Negatiu	Positiu	
Real	Negatiu	97	1	98
	Positiu	1	1	2

Figura 42. Matriu de confusió

$$Accuracy = \frac{97 + 1}{100} = 98\%$$

La precisió és del 0.98 o 98%, és a dir, 98 prediccions correctes de 100 exemples totals. A primera vista es podria pensar que el classificador està fent un bon treball a l'hora d'identificar els suspesos. Es realitza un anàlisi més detallat per conèixer millor el rendiment del model.

Dels 100 exemples de notes, 98 són aprovats (97 TN i 1 FP) i 2 són suspesos (1 TP i 1 FN).

- Dels 98 aprovats, el model identifica correctament 97 aprovats, un resultat molt bo.
- No obstant, dels 2 aprovats, el model només identifica correctament 1 suspès, és a dir, la meitat. És un resultat força preocupant.

Aquest model presentaria la mateixa precisió que un que classifiqués totes les instàncies com a aprovades (98/100). Això es coneix com la paradoxa de la precisió.

A més, la precisió no té en compte el cost associat dels FN i FP. En el nostre cas ens hem de preguntar: quin és el cost associat de predir l'aprovat d'un alumne quan en realitat suspendrà l'assignatura? O bé: quin és el cost associat de predir el suspès d'un alumne quan en realitat aprovarà l'assignatura?

Així dons, no ens podem basar només amb la mesura de precisió, s'hauran d'utilitzar altres paràmetres com l'exactitud, la sensibilitat i el valor F1 per tenir en compte aquests costos associats.

Paràmetres d'interès

La predicció del suspès d'una assignatura per un alumne pot ser una informació útil per el professorat. Si mitjançant les notes de la fase inicial es preveu que aquest alumne suspendrà una assignatura, els professors poden fer quelcom el respecte així com proporcionar-los més documentació o bé portar un seguiment més constant de l'alumne. És per això, que el cost dels

falsos negatius és a dir, aquells alumnes que es preveuen com a aprovats però en realitat suspenen, és molt elevat. L'estudi es basarà sobretot en el paràmetre *Recall.*

Els falsos positius per altra banda, presenten un cost associat menor. Es considera que ajudar aquells alumnes que es prediuen com a suspesos és un "mal menor". Els alumnes poden utilitzar o no aquesta ajuda que se'ls hi ha ofert o bé utilitzar-la per treure millors notes. No obstant, també interessa minimitzar aquests falsos positius doncs no deixen de ser prediccions errònies i el seu augment suposa una disminució dels suspesos reals. L'estudi tindrà en compte, en menor mesura, el paràmetre *Precision.*

Finalment, s'estudiarà el valor F1 doncs és una mitjana dels paràmetres *Recall* i *Precision* i s'utilitza quan els falsos negatius i falsos positius presenten costos associats, com en el present cas.

El paràmetre *Accuracy* només s'estudiarà en cas d'una distribució de classes equilibrada.

4.3. Predicció mitjançant Regressió Logística

S'utilitzarà la validació creuada amb K=3. A continuació es mostra el procediment seguit:

1. Es barreja aleatòriament el *Dataset*
2. Es divideix aleatòriament el conjunt de dades en 3 seccions iguals.
3. Per a cada secció K [1:3]:
    a. Es tracta com a *Test set* o *Hold out set*
    b. Es pren la resta de seccions com a *Training set*
    c. S'ajusta el model en el *Training* set
    d. S'avalua el model en el *Test set* mitjançant una mètrica de rendiment (*Accuracy, F1...)*
4. Es fa la mitjana de les 3 mètriques de cada iteració.

Figura 43. Exemple de validació creuada per K=5

Per a la divisió de les dades en seccions, s'utilitzarà la funció *StratifiedKFold* la qual retornarà seccions amb el mateix percentatge d'instàncies de cada classe.

S'avaluaran els resultats de les assignatures del Q3 en funció dels paràmetres *Accuracy, Recall i F1 Value* i es compararan els resultats entre els *DataFrames.*

La matriu resultant per a cada assignatura és la suma de totes les matrius obtingudes per cada secció K. Les mètriques, per altre banda, són la mitjana de les iteracions. És a dir, no surten d'aplicar les seves respectives fórmules a partir de la matriu, tot i que en alguns casos, els resultats podrien ser semblants.

Predicció a l'assignatura d'Electromagnetisme

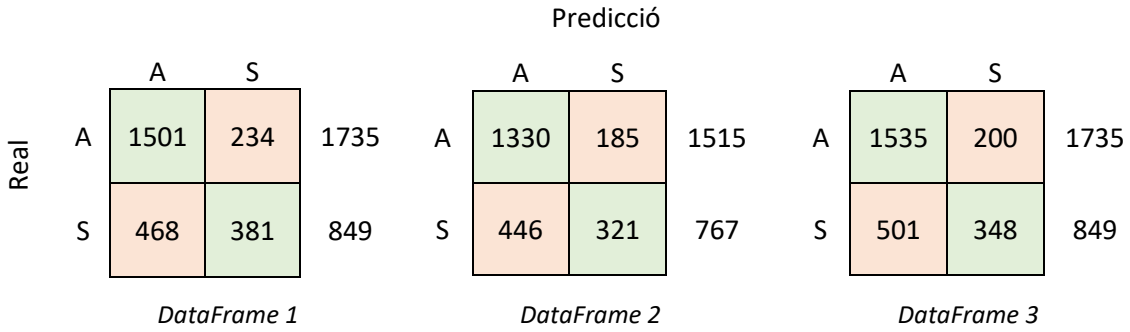


Figura 44. Matrius de confusió de l'assignatura d'electromagnetisme

	Accuracy	Precision	Recall	F1-Score
Dataframe 1	0,7283	0,6256	0,4488	0,5213
Dataframe 2	0,7235	0,6350	0,4185	0,5041
Dataframe 3	0,7287	0,6427	0,4099	0,4988

Taula 11. Taula de resultats de l'assignatura d'electromagnetisme

No s'observa gaire diferència de resultats entre els *DataFrames.* El paràmetre *Precision* és més elevat que el *Recall* i el paràmetre *F1 Score* es troba entre els dos valors.

Predicció a l'assignatura de Mètodes Numèrics

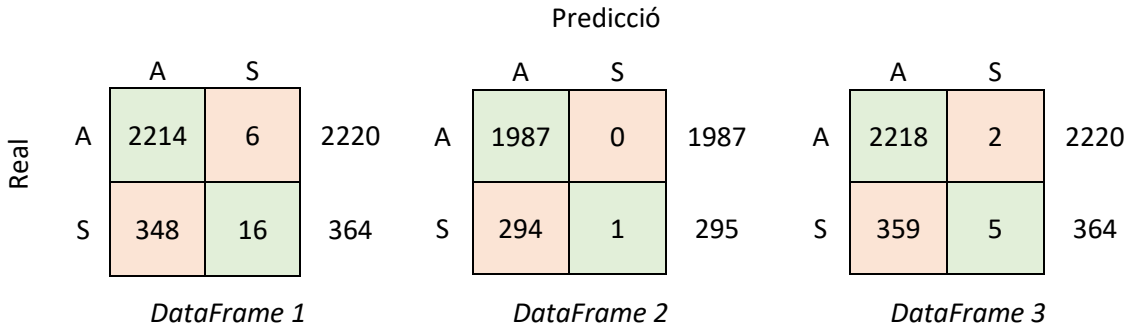


Figura 45. Matrius de confusió de l'assignatura de Mètodes Numèrics

	Accuracy	Precision	Recall	F1-Score
Dataframe 1	0,8630	0,7315	0,0439	0,0824
Dataframe 2	0,8712	0,3333	0,0034	0,0067
Dataframe 3	0,8603	0,6667	0,0137	0,0269

Taula 12. Taula de resultats de l'assignatura de Mètodes Numèrics

Es troben diferències entre els tres *DataFrames.* El primer d'ells obté els millors resultats; l'addicció de noves variables com la nota de selectivitat o el nombre de repeticions no aporten millores.

S'observa que el paràmetre *Recall* i *F1* és molt baix en tots els casos.

Predicció a l'assignatura de Materials

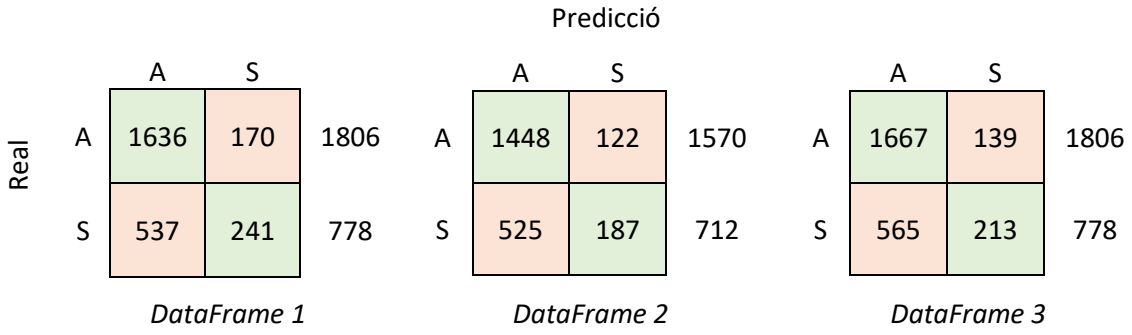


Figura 46. Matrius de confusió de l'assignatura de Materials

	Accuracy	Precision	Recall	F1-Score
Dataframe 1	0,7264	0,5884	0,3098	0,4055
Dataframe 2	0,7165	0,6073	0,2627	0,3658
Dataframe 3	0,7276	0,6084	0,2738	0,3772

Taula 13. Taula de resultats de l'assignatura de Materials

En general, els models presenten un paràmetres bastant similars entre ells.

El primer *DataFrame* és qui presenta els millors valors dels paràmetres *Recall* i *F1-Score.* En aquest cas, l'addició de noves dades tampoc comporta millores.

Predicció a l'assignatura de Equacions diferencials

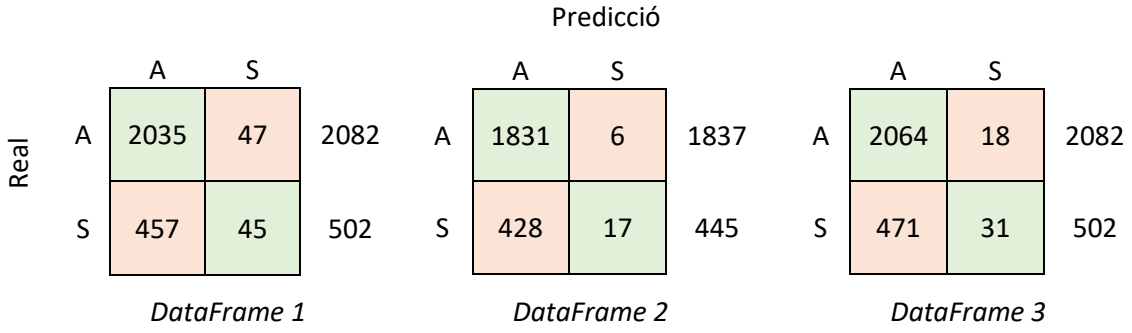


Figura 47 Matrius de confusió de l'assignatura d'Equacions diferencials

	Accuracy	Precision	Recall	F1-Score
Dataframe 1	0,8049	0,5004	0,0896	0,1494
Dataframe 2	0,8098	0,6820	0,0383	0,0717
Dataframe 3	0,8108	0,6505	0,0617	0,1108

Taula 14. Taula de resultats de l'assignatura d'Equacions diferencials

El primer *Dataframe* obté els millors resultats tot i que paràmetre *Recall* i *F1* és molt baix en tots els casos.

Predicció a l'assignatura de Informàtica

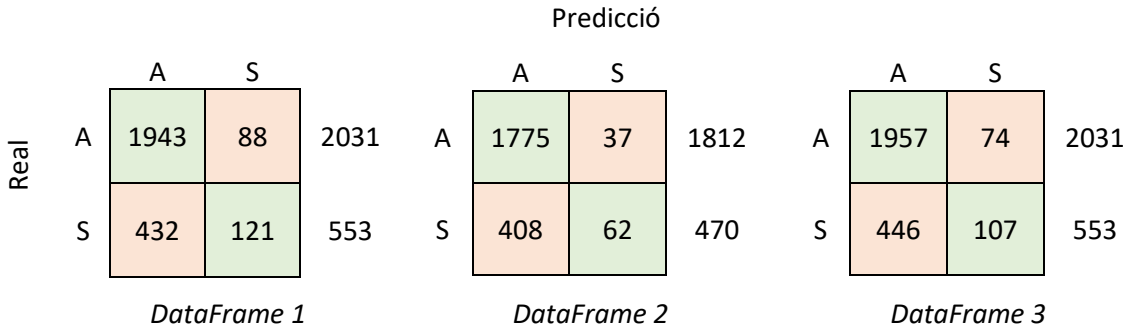


Figura 48. Matrius de confusió de l'assignatura d'Informàtica

	Accuracy	Precision	Recall	F1-Score
Dataframe 1	0,7988	0,5753	0,2187	0,3167
Dataframe 2	0,8050	0,6617	0,1320	0,2152
Dataframe 3	0,7988	0,5891	0,1934	0,2911

Taula 15. Taula de resultats de l'assignatura d'Informàtica

Els models presenten un paràmetres bastant similars entre ells encara que el primer *DataFrame* és qui presenta els millors valors dels paràmetres *Recall* i *F1-Score.*

Predicció a l'assignatura de Mecànica

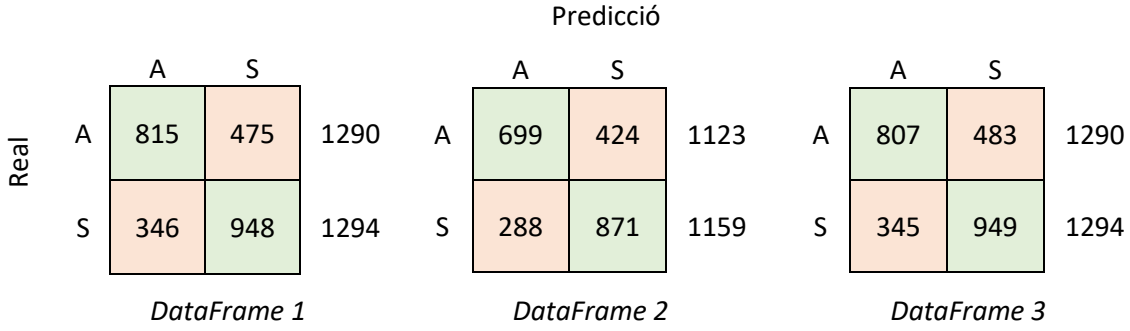


Figura 49. Matrius de confusió de l'assignatura de Mecànica

	Accuracy	Precision	Recall	F1-Score
Dataframe 1	0,6822	0,6662	0,7326	0,6975
Dataframe 2	0,6880	0,6729	0,7515	0,7100
Dataframe 3	0,6796	0,6625	0,7334	0,6957

Taula 16. Taula de resultats de l'assignatura de Mecànica

Tots els paràmetres presenten valors molt similars en els tres casos. El paràmetre *Recall* assoleix valors més elevats que el paràmetre *Precision.*

Conclusions

Les assignatures de *Mètodes Numèrics, Equacions diferencials i Informàtica* presenten un mateix patró en els resultats. El paràmetre *Recall* obté valors extremadament petits; això es pot traduir en que el model no identifica correctament els suspesos. Existeixen un gran nombre d'aquests que el model prediu com a aprovats, coneguts com falsos negatius. Aquestes prediccions errònies són els que es volen minimitzar al màxim, obtenint un *Recall* elevat. És per això que el model de regressió logística no servirà per a la modelització d'aquestes assignatures.

Una possible raó per la qual el model presenta aquesta quantitat de falsos aprovats és a causa de la distribució desequilibrada de classes. El model prediu erròniament un gran nombre de suspesos (*Recall* baix) però encerta la majoria dels aprovats (*Precision* elevat). Els mètodes d'aprenentatge automàtic no funcionen bé sobre conjunts de dades desequilibrats, donant lloc a un biaix a favor de la classe majoritària. El biaix són els supòsits realitzats pel model sobre la forma de la funció objectiu. Durant la fase d'entrenament, la classe minoritària contribueix menys sobre la funció objectiu doncs presenta menys exemples per entrenar. El model és més sensible a la detecció dels aprovats, classe majoritària, i menys sensible a la classe minoritària, els suspesos. Una solució seria equilibrar la distribució de classes i veure si milloren els resultats.

Les assignatures d'*Electromagnetisme i Materials* també presenten una distribució de dades desequilibrada. Es recorda que en aquests casos, el paràmetre *Accuracy* no serà gaire representatiu. El paràmetre *Recall* presenta valors inferiors que el paràmetre *Precision* i el valor *F1* es troba entre ambdós. El paràmetre *Precision* presenta un valor al voltant de 0,6: es podria afirmar que el model identifica bastant bé els aprovats. A partir de les matrius de confusió, s'observa que el nombre d'aprovats predits erròniament és mínim. No obstant, no prediu correctament la classe minoritària, els suspesos, que és el principal interès. Com en el cas anterior, una solució podria ser tornar a modelar amb una distribució equilibrada de classes.

Per últim, l'assignatura de *Mecànica* és la única que presenta una distribució de classes equilibrada. Alguns la consideren l'assignatura més difícil de la carrera i això es pot veure reflectit en que hi ha aproximadament el mateix nombre de suspesos i aprovats d'alumnes que cursen aquesta assignatura per primera vegada. Fins i tot, el nombre de suspesos es lleugerament superior. En aquest cas, el paràmetre *Accuracy* serà representatiu.

Tots els *Dataframes* presenten valors dels paràmetres molt similars entre ells. El paràmetre *Accuracy* presenta un valor d'aproximadament 0,68, és a dir, el model identifica correctament un 68% dels casos d'entre totes les dades. Les dades compleixen amb els supòsits i s'ajusten a la funció de la regressió logística de manera acurada. Això podria ser a causa de la distribució de dades equilibrada.

El paràmetre *Recall* presenta valors superiors al paràmetre *Precision* i el valor *F1* es troba entre els dos valors. Es podria afirmar que la predicció del suspès és més fiable que la del aprovat. El paràmetre *Recall* assoleix valors d'aproximadament 0,74, el millor valor assolit fins ara.

4.4. Predicció mitjançant K-Nearest Neighbors

En el cas de KNN es volen utilitzar les tècniques de validació no només per trobar el millor model, sinó també per optimitzar l'híper-paràmetre k. A l'apartat 3.3 es va comentar com dividir el conjunt de dades en tres subconjunts, entrenament, proves i validació, per tal de trobar l'híper-paràmetre k. El conjunt de validació actuava com un conjunt de proves i es realitzava una validació creuada interna amb el conjunt d'entrenament. Això es coneix com a *Nested Cross Validation*. Una vegada s'ha trobat el valor de k més adequat, es procedeix a construir el model i es valida mitjançant la validació *K-Cross Validation* externa.

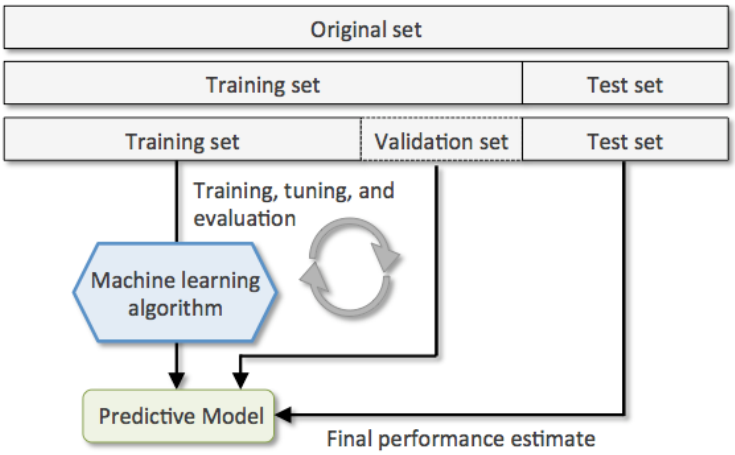


Figura 50. Procés d'optimització de paràmetres

S'utilitzarà la Validació creuada conjuntament tant a la selecció d'híper-paràmetre com per la validació amb una K de validació K=3 en ambdós casos: 3x3 *Nested CV.*

1. Es barreja aleatòriament el *Dataset*
2. Es divideix el conjunt de dades en 3 seccions iguals (M=3)[11]
3. Per a cada secció/iteració M [1:3] es durà a terme una validació creuada interna:
    a. Per cada k del KNN [1,n]:
        i. Dividim el *Training set* en K_{CV}=3 seccions. Per a cada iteració:
            1. S'ajusta el model en el *CV-Train*
            2. S'avalua el model en el *CV-Validation* i mitjançant una mètrica de rendiment.
        ii. Es fa la mitjana de les 3 mètriques de rendiment.
    b. Per cada k una mètrica de validació associada.
    c. Es seleccionen les 5 millors k segons la mètrica.
4. Per cada secció M es té els 5 millors valors associats a una k (una matriu M x k)
    a. Es selecciona manualment el millor valor de k, k*
5. Es realitza la Validació Creuada externa amb M=3 com s'ha explicat a la Regressió logística.

[11] Es designarà M a les seccions de la validació creuada externa per no confondre-ho amb les seccions K de la validació interna.

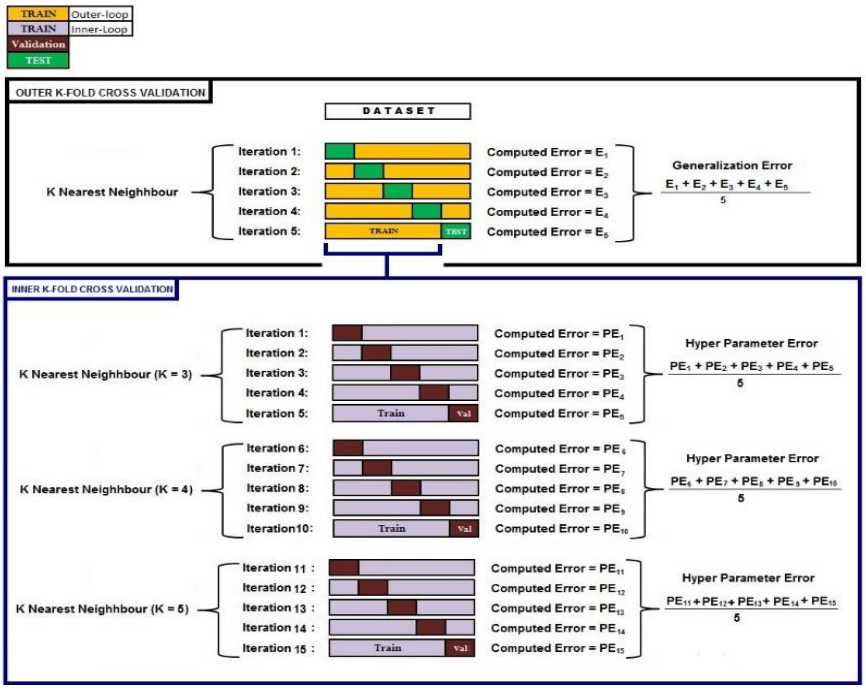


Figura 51. Exemple de 5x5 Nested CV

Tant per la validació creuada interna com externa, s'utilitzarà la funció *StratifiedKFold* la qual retornarà seccions amb el mateix percentatge d'instàncies de cada classe.

Per dur a terme la validació interna explicada al pas 3, s'utilitzarà la funció *sklearn.model_selection.GridSearchCV* per a cada secció M. Aquesta funció cerca els valors dels híper-paràmetres desitjats per a un estimador. Els seus paràmetres a especificar són:

- ***Estimator:*** KNN mitjançant el mòdul *sklearn.neighbors.KNeighborsClassifier()*

- ***Param_grid:*** El rang de valors de l'híper-paràmetre k. Es sap que k pot prendre valors entre [1:n] on n és el nombre d'instàncies del conjunt d'entrenament. No obstant, avaluar el conjunt de validació per totes aquestes k comporta un compost computacional elevat. Existeix una regla general per el paràmetre k la qual afirma que el seu valor s'aproxima a $\sqrt{n}$. Sabent això, s'establirà aquest valor com a límit del rang de valors de k multiplicat per dos per donar una mica més de marge. Així doncs, per al procés de cerca d'aquest híper-paràmetre, k podrà prendre valors d'entre 1 i $2\sqrt{n}$.

- ***Scoring:*** Les mètriques de rendiment que es volen per avaluar el conjunt de validació. Aquesta mètrica serà la responsable de la tria del valor de k, doncs es prendran els cinc millors valors d'aquesta associats a una k. Tornant a l'apartat 3.3 on s'explicava el procés intern d'optimització del paràmetre k, es prenia com a mètrica l'error de validació associat i s'escollia el valor mínim d'aquest. Ara bé, aquesta mètrica pot variar segons l'estudi i context. En aquest cas, es designarà el paràmetre a *Recall* ja que és la mètrica que té en compte el nombre de falsos negatius, preocupació principal de l'estudi. S'escolliran els cinc millors valors més elevats del paràmetre.

- ***Cv***: El nombre de seccions K per a la validació creuada interna. Es prendrà K=3.

S'utilitzarà l'atribut *cv_results* de la funció *GridSearchCV*, el qual retorna un diccionari[12] amb les claus com un *string*[13] del resultat obtingut i els valors una llista de valors associats per a cada k. Aquests valors venen ordenats segons el valor creixent de k. Es seleccionarà la clau *mean_test_recall* i retornarà una llista de la mitjana de la mètrica *Recall* avaluada en els conjunts de validació. A continuació, es crea una llista dels possibles valors dins els rang establert. Es combinen ambdues llistes per crear una nova on els seus valors són una llista de [valor, k] de manera que cada valor del paràmetre *Recall* se li associa la seva k corresponent. S'ordena la llista en ordre descendent segons el paràmetre *Recall*, essent el primer valor el millor associat a la millor k. Finalment, es prenen els 5 primers valors d'aquesta llista.

Per cada assignatura i *Dataframe*, es mostra la matriu associada Mxk del pas 4. Es selecciona manualment el valor k de manera que sigui el millor valor d'entre les tres seccions M. No es pot prendre k=1 ja que es produiria *Overfitting* ni k=nº de mostres del conjunt de validació ja que es produiria l'efecte contrari, *Underfitting.*

Predicció a l'assignatura d'Electromagnetisme

- *Dataframe 1*

M=1	0,4823; k=1	0,42931; k=3	0,4258; k=5	0,4046; k=7	0,3975; k=9
M=2	0,4805; k=1	0,4611; k=3	0,4452; k=13	0,4435; k=5	0,4346; k=17
M=3	0,4753; k=1	0,4258; k=5	0,4258; k=3	0,3922; k=7	0,3887; k=9

Taula 17. Taula de valors de k i Recall de l'assignatura d'Electromagnetisme, Df1

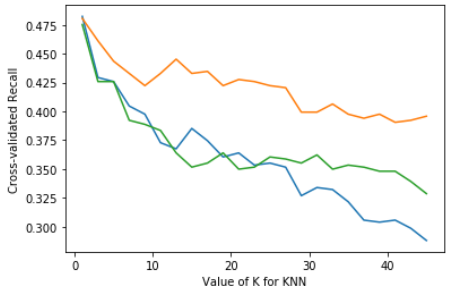


Figura 52. Gràfic paràmetre K-Recall d'Electromagnetisme, Df 1

- *Dataframe 2*

M=1	0,4873; k=1	0,4756; k=5	0,4736; k=15	0,4677; k=11	0,4658; k=19
M=2	0,4501; k=1	0,4266; k=5	0,4129; k=3	0,4090; k=7	0,4051; k=9
M=3	0,5019; k=3	0,4941; k=1	0,4609; k=5	0,4570; k=7	0,4238; k=9

Taula 18. Taula de valors de k i Recall de l'assignatura d'Electromagnetisme, Df2

[12]Estructura de dades que pot emmagatzemar qualsevol tipus de dades. Permeten identificar cada element a partir d'una clau.

[13] En codi *Python*, format d'element que pot incorporar components tant alfabètics com numèrics

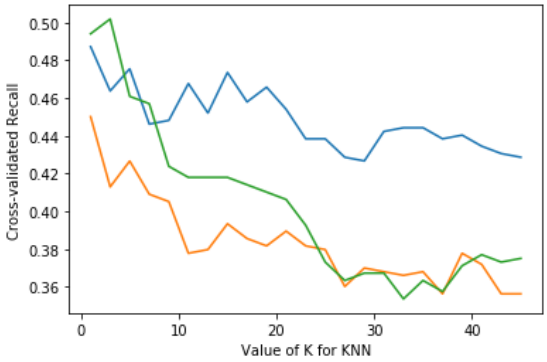


Figura 53. Gràfic paràmetre K-Recall d'Electromagnetisme, Df2

- *Dataframe 3*

M=1	0,4823; k=1	0,4293; k=3	0,4258; k=5	0,4046; k=7	0,3975; k=9
M=2	0,4788; k=1	0,4611; k=3	0,4452; k=13	0,4435; k=5	0,4346; k=17
M=3	0,4753; k=1	0,4258; k=5	0,4258; k=3	0,3922; k=7	0,3887; k=9

Taula 19. Taula de valors de k i Recall de l'assignatura d'Electromagnetisme, Df3

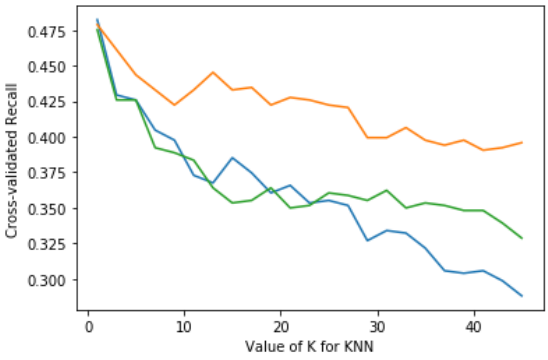


Figura 54. Gràfic paràmetre K-Recall d'Electromagnetisme, Df 3

Una vegada establert el valor k*, es procedeix a realitzar la validació creuada per M=3 de manera que s'obtenen els següents resultats:

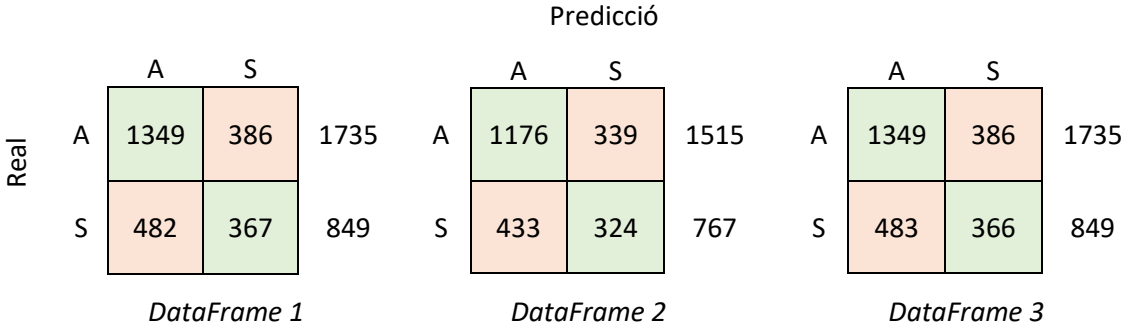


Figura 55. Matrius de confusió de l'assignatura d'Electromagnetisme

	k	ICV-Recall	Accuracy	Precision	Recall	F1-Score
Dataframe 1	3	0,4387	0,6648	0,4895	0,4276	0,4555
Dataframe 2	5	0,4544	0,6722	0,5152	0,4315	0,4695
Dataframe 3	3	0,4387	0,6648	0,4895	0,4276	0,4555

Taula 20. Taula de valors de l'assignatura d'Electromagnetisme

No s'observa gaire diferència de resultats entre els *DataFrames.* El paràmetre *Precision* és més elevat que el *Recall* i el paràmetre *F1 Score* es troba entre els dos valors.

El valor de l'hiper-paràmetre k varia per cada *Dataframe.* Això té sentit ja que corresponen a conjunts de dades diferents amb diferents sensibilitats al valor k.

El valor del paràmetre *Recall* obtingut mitjançant la validació creuada interna, *Inner Cross-Validation Recall,* és lleugerament superior a l'obtingut mitjançant el *Test set* de la validació creuada externa. Això es tradueix en que alhora de la selecció del híper-paràmetre hi ha hagut un problema d'*Overfitting.* El model s'ajusta molt bé al conjunt de dades del *Training set* (*Inner Cross Validation)* donant estimacions de paràmetres altes però no presenta un bon rendiment sobre les noves dades. Una possible solució a aquest problema seria canviar el mètode de validació, afegint més seccions com per exemple K_{CV}=5.

Predicció a l'assignatura de Mètodes Numèrics

- *Dataframe 1*

M=1	0,2232, k=1	0,1448, k=3	0,1156, k=5	0,0951, k=7	0,0744, k=9
M=2	0,1605, k=1	0,1111, k=3	0,0699, k=5	0,0617, k=7	0,0494, k=9
M=3	0,2098, k=1	0,0946, k=3	0,0823, k=5	0,0782, k=7	0,0452, k=9

Taula 21. Taula de valors de k i Recall de l'assignatura de Mètodes numèrics, Df1

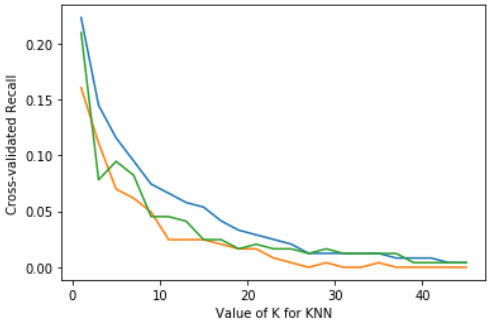


Figura 56. Gràfic paràmetre K-Recall de Mètodes numèrics, Df 1

- *Dataframe 2*

M=1	0,1633; k=1	0,1073; k=3	0,0716; k=7	0,0664; k=5	0,0410; k=11
M=2	0,1724; k=1	0,1523; k=3	0,1013; k=7	0,0911; k=5	0,0761; k=9
M=3	0,2794; k=1	0,1219; k=3	0,0813; k=5	0,0558; k=7	0,0405; k=9

Taula 22. Taula de valors de k i Recall de l'assignatura de Mètodes numèrics, Df2

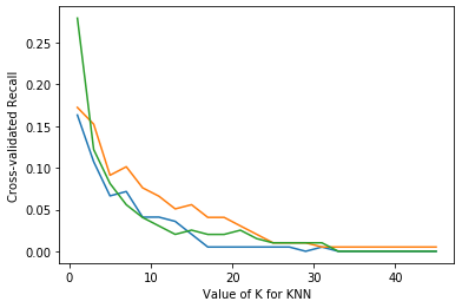


Figura 57. Gràfic paràmetre K-Recall de Mètodes numèrics, Df 2

- *Dataframe 3*

M=1	0,1633; k=1	0,1073; k=3	0,0716; k=5	0,0664; k=7	0,0410; k=9
M=2	0,1724; k=1	0,1523; k=3	0,1013; k=5	0,0911; k=7	0,0761; k=9
M=3	0,2794; k=1	0,1219; k=5	0,0813; k=7	0,0558; k=3	0,0405; k=9

Taula 23. Taula de valors de k i Recall de l'assignatura de Mètodes numèrics, Df3

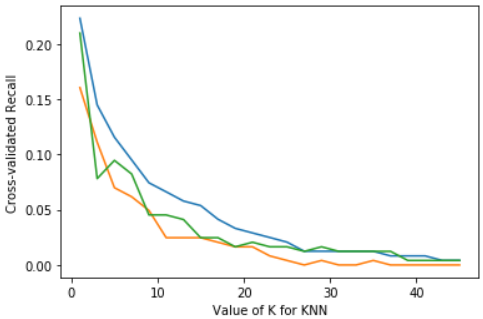


Figura 58. Gràfic paràmetre K-Recall de Mètodes numèrics, Df 3

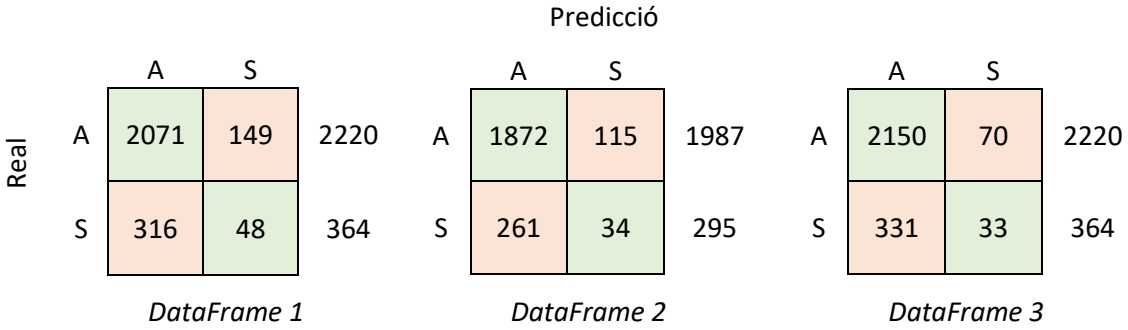


Figura 59. Matrius de confusió de l'assignatura de Mètodes numèrics

	k	ICV-Recall	Accuracy	Precision	Recall	F1-Score
Dataframe 1	3	0,1168	0,8231	0,2461	0,1208	0,1609
Dataframe 2	3	0,1272	0,8431	0,2369	0,0915	0,1303
Dataframe 3	5	0,0983	0,8460	0,3292	0,0879	0,1387

Taula 24. Taula de valors de l'assignatura de Mètodes numèrics

El primer *Dataframe* obté els millors resultats; l'addicció de noves variables no aporten millores.

El valor del paràmetre *Recall* obtingut mitjançant la validació creuada interna és semblant al obtingut mitjançant el *Test set.* Això significa que el nombre de seccions K_{CV} per escollir el paràmetre k ha funcionat correctament.

Predicció a l'assignatura de Materials

- *Dataframe 1*

M=1	0,3939; k=1	0,3146; k=3	0,3108; k=5	0,2780; k=7	0,2510; k=9
M=2	0,3892; k=1	0,3564; k=3	0,3217; k=5	0,2948; k=7	0,2813; k=9
M=3	0,4104; k=1	0,3680; k=3	0,3507; k=5	0,3372; k=7	0,3063; k=9

Taula 25. Taula de valors de k i Recall de l'assignatura de Materials, Df1

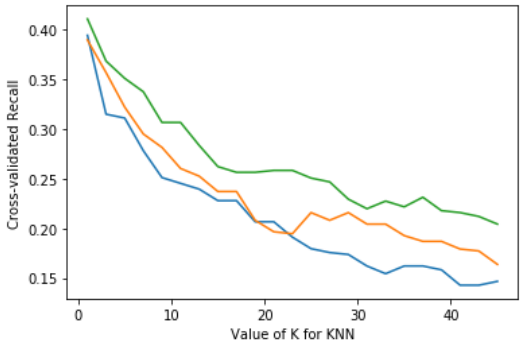


Figura 60. Gràfic paràmetre K-Recall de Materials, Df1

- *Dataframe 2*

M=1	0,4008; k=1	0,3798; k=3	0,3481; k=11	0,3334; k=7	0,3334; k=5
M=2	0,3852; k=1	0,3832; k=3	0,3326; k=5	0,3073; k=7	0,2863; k=9
M=3	0,4547; k=1	0,3936; k=3	0,3705; k=7	0,3579; k=5	0,3369; k=9

Taula 26. Taula de valors de k i Recall de l'assignatura de Materials, Df2

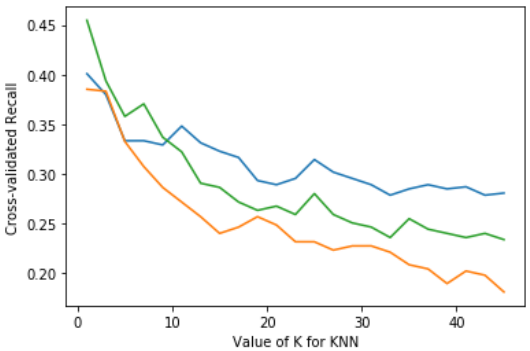


Figura 61. Gràfic paràmetre K-Recall de Materials, Df2

- *Dataframe 3*

M=1	0,3939; k=1	0,3127; k=3	0,3108; k=5	0,2780; k=7	0,2510; k=9
M=2	0,3892; k=1	0,3564; k=3	0,3217; k=5	0,2948; k=7	0,2813; k=9
M=3	0,4104; k=1	0,3680; k=3	0,3507; k=5	0,3372; k=7	0,3063; k=9

Taula 27. Taula de valors de k i Recall de l'assignatura de Materials, Df3

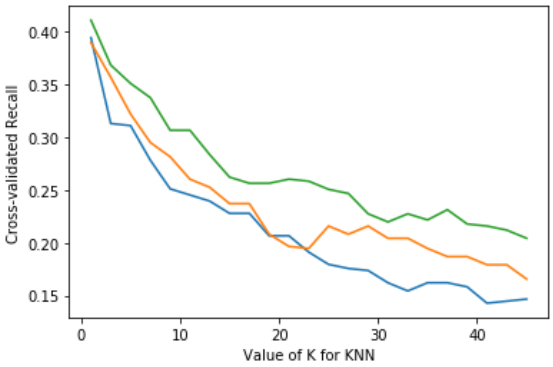


Figura 62. Gràfic paràmetre K-Recall de Materials, Df3

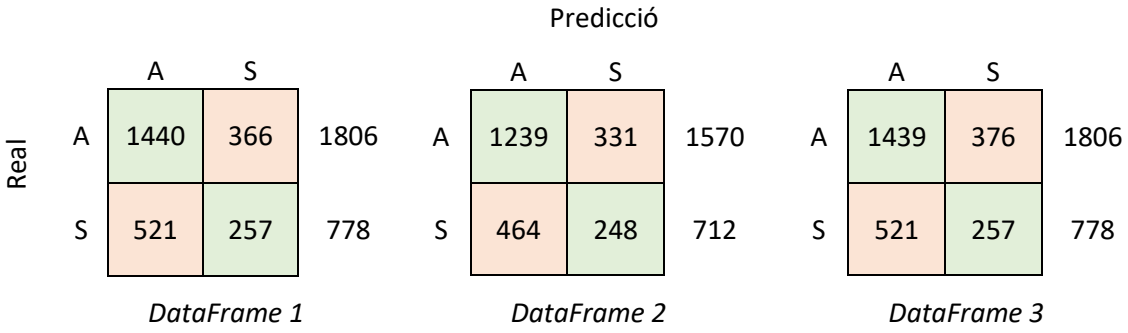


Figura 63. Matrius de confusió de l'assignatura de Materials

	k	CV-Recall	Accuracy	Precision	Recall	F1-Score
Dataframe 1	3	0,3463	0,6660	0,4320	0,3470	0,3849
Dataframe 2	3	0,3855	0,6661	0,4570	0,3638	0,4046
Dataframe 3	3	0,3457	0,6656	0,4312	0,3470	0,3846

Taula 28. Taula de valors de l'assignatura de Materials

Els models presenten un paràmetres bastant similars entre ells. El segon *DataFrame* és el que presenta valors dels paràmetres *Recall* i *F1-Score* mínimament superiors.

En general, el valor del paràmetre *Recall* obtingut mitjançant la validació creuada interna és similar al obtingut mitjançant la validació externa. Això significa que el nombre de seccions K_{CV} per escollir el paràmetre k ha funcionat correctament.

Predicció a l'assignatura d'Equacions diferencials

- *Dataframe 1*

M=1	0,2725, k=1	0,2006, k=3	0,1377, k=5	0,1226, k=7	0,0988, k=9
M=2	0,2986, k=1	0,2239, k=3	0,1582, k=5	0,1314, k=7	0,1283, k=11
M=3	0,2955, k=1	0,2448, k=3	0,1881, k=5	0,1642, k=7	0,1403, k=9

Taula 29. Taula de valors de k i Recall de l'assignatura d'Equacions diferencials, Df1

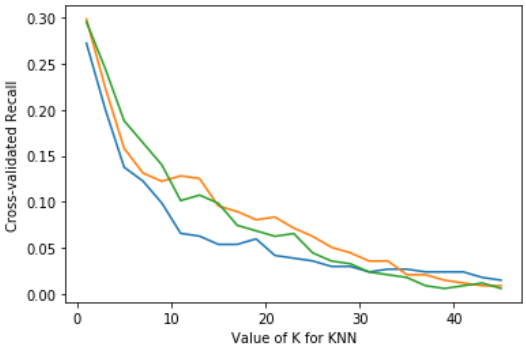


Figura 64. Gràfic paràmetre K-Recall d'Equacions diferencials, Df 1

- *Dataframe 2*

M=1	0,2870; k=1	0,1824; k=3	0,1419; k=5	0,0879; k=9	0,0879; k=7
M=2	0,2862; k=1	0,2088; k=3	0,1751; k=5	0,1583; k=7	0,1482; k=9
M=3	0,2694; k=1	0,1987; k=3	0,1616; k=5	0,1515; k=7	0,1448; k=11

Taula 30. Taula de valors de k i Recall de l'assignatura d'Equacions diferencials, Df2

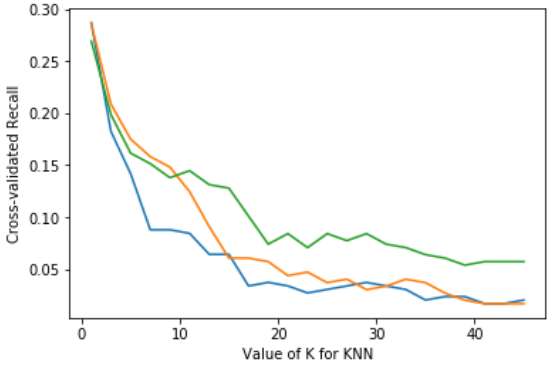


Figura 65. Gràfic paràmetre K-Recall d'Equacions diferencials, Df 2

- *Dataframe 3*

M=1	0,2725; k=1	0,2006; k=3	0,1377; k=5	0,1226; k=7	0,0988; k=9
M=2	0,2986; k=1	0,2239; k=3	0,1582; k=5	0,1314; k=7	0,1283; k=11
M=3	0,2955; k=1	0,2448; k=3	0,1881; k=5	0,1642; k=7	0,1403; k=9

Taula 31. Taula de valors de k i Recall de l'assignatura d'Equacions diferencials, Df3

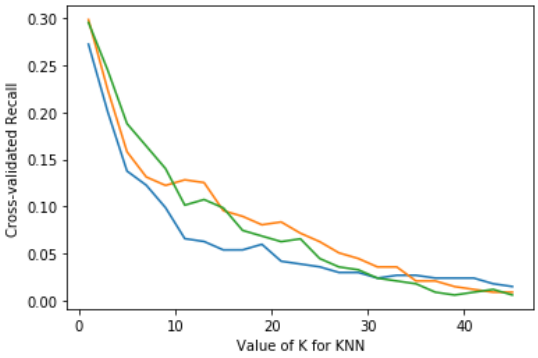


Figura 66. Gràfic paràmetre K-Recall d'Equacions diferencials, Df 3

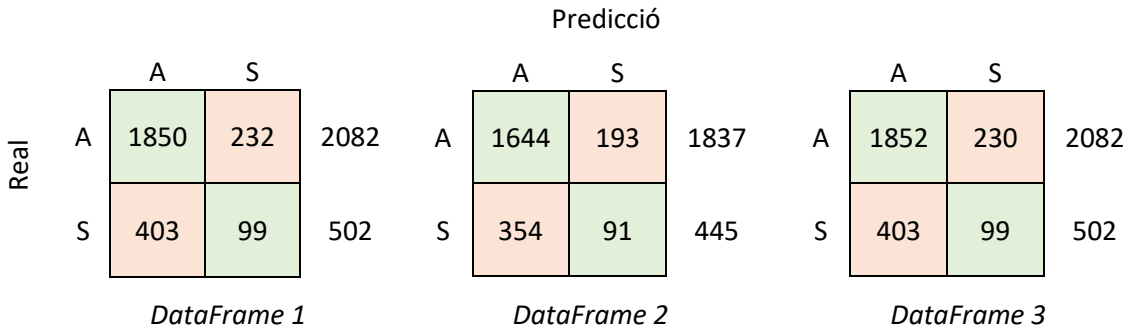


Figura 67. Matrius de confusió de l'assignatura d'Equacions diferencials

	k	CV-Recall	Accuracy	Precision	Recall	F1-Score
Dataframe 1	3	0,2231	0,7647	0,3301	0,2032	0,2514
Dataframe 2	3	0,1966	0,7774	0,3848	0,2314	0,2884
Dataframe 3	3	0,2231	0,7643	0,3290	0,2032	0,2511

Taula 32. Taula de valors de l'assignatura d'Equacions diferencials

Els models presenten un paràmetres bastant similars entre ells i a la vegada molt baixos. El segon *DataFrame* és el que presenta valors dels paràmetres *Recall* i *F1-Score* mínimament superiors.

El valor del paràmetre *Recall* obtingut mitjançant la validació creuada interna és semblant al obtingut mitjançant el *Test set.* No es té cap problema d'*Overfitting.*

Predicció a l'assignatura d'Informàtica

- *Dataframe 1*

M=1	0,3261, k=1	0,2662, k=3	0,2391, k=5	0,1956, k=7	0,1876, k=9
M=2	0,3333, k=1	0,2927, k=3	0,2683, k=5	0,2683, k=7	0,2493, k=9
M=3	0,2981, k=1	0,2683, k=3	0,2439, k=5	0,2303, k=7	0,2276, k=9

Taula 33. Taula de valors de k i Recall de l'assignatura d'Informàtica, Df1

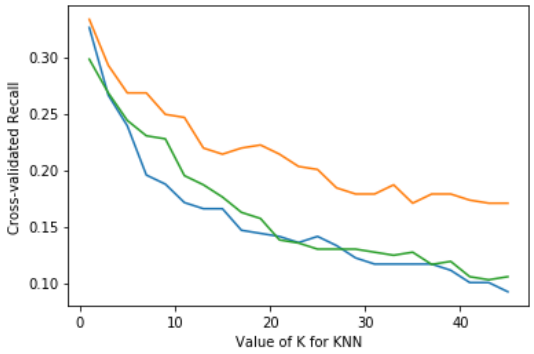


Figura 68. Gràfic paràmetre K-Recall d'Informàtica, Df 1

- *Dataframe 2*

M=1	0,3162; k=1	0,2746; k=3	0,2523; k=5	0,2268; k=7	0,2203; k=13
M=2	0,3289; k=1	0,2747; k=3	0,2651; k=5	0,2172; k=7	0,1980; k=9
M=3	0,3057; k=1	0,2770; k=3	0,2516; k=5	0,2388; k=7	0,2357; k=9

Taula 34. Taula de valors de k i Recall de l'assignatura d'Informàtica, Df2

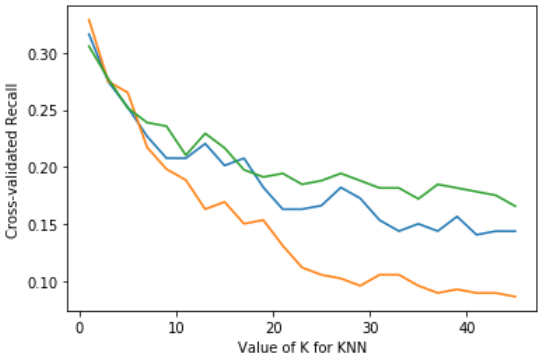


Figura 69. Gràfic paràmetre K-Recall d'Informàtica, Df 2

- *Dataframe 3*

M=1	0,3261; k=1	0,2662; k=3	0,2419; k=5	0,1929; k=7	0,1876; k=9
M=2	0,3334; k=1	0,2927; k=3	0,2683; k=5	0,2682; k=7	0,2493; k=9
M=3	0,2981; k=1	0,2683; k=3	0,2439; k=5	0,2303; k=7	0,2276; k=9

Taula 35. Taula de valors de k i Recall de l'assignatura d'Informàtica, Df3

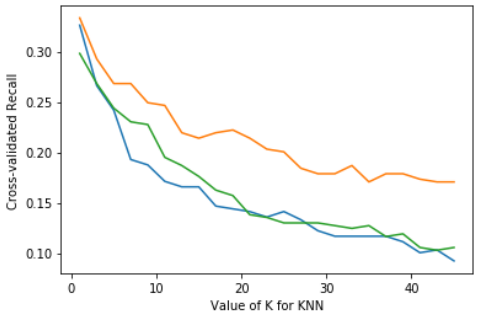


Figura 70. Gràfic paràmetre K-Recall d'Informàtica, Df 3

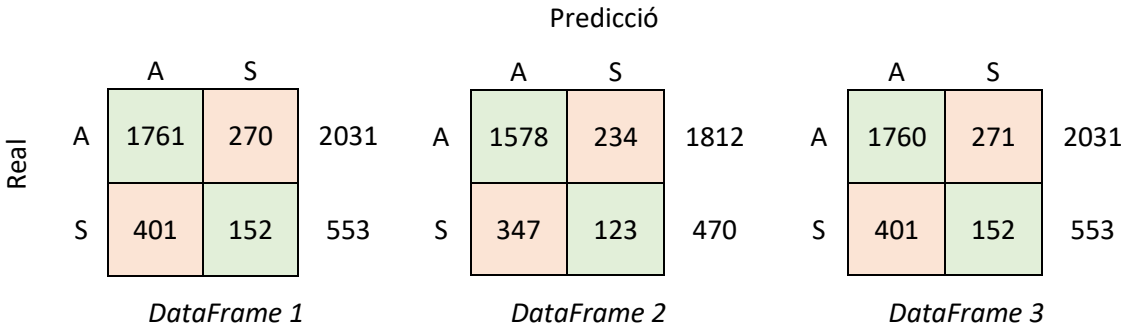


Figura 71. Matrius de confusió de l'assignatura d'Informàtica

	k	CV-Recall	Accuracy	Precision	Recall	F1-Score
Dataframe 1	3	0,2757	0,7535	0,3936	0,2892	0,3327
Dataframe 2	3	0,2754	0,7651	0,3985	0,2638	0,3171
Dataframe 3	3	0,2757	0,7535	0,3936	0,2892	0,3327

Taula 36. Taula de valors de l'assignatura d'Informàtica

El primer i tercer *Dataframe* presenten resultats idèntics. El segon *Dataframe* assumeix valor de *Recall i F-Score* inferiors.

Generalment, el valor del paràmetre *Recall* obtingut mitjançant la validació creuada interna és similar al obtingut mitjançant la validació externa. El nombre de seccions K_{CV} per escollir el paràmetre k ha estat correcte.

Predicció a l'assignatura de Mecànica

- *Dataframe 1*

M=1	0,7808; k=41	0,7784; k=43	0,7761; k=39	0,7738; k=45	0,7738; k=37
M=2	0,8053; k=81	0,8030; k=77	0,8007; k=79	0,7984; k=75	0,7949; k=73
M=3	0,7961; k=75	0,7937; k=79	0,7926; k=77	0,7879; k=81	0,7868; k=73

Taula 37. Taula de valors de k i Recall de l'assignatura de Mecànica, Df1

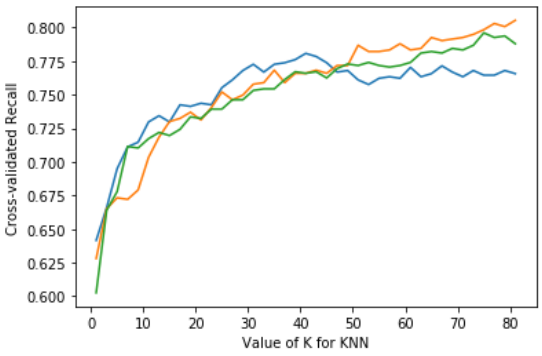


Figura 72. Gràfic paràmetre K-Recall de Mecànica, Df 1

- *Dataframe 2*

M=1	0,7992; k=77	0,7940; k=75	0,7914; k=67	0,7914; k=73	0,7902; k=43
M=2	0,8047; k=77	0,8034; k=71	0,8021; k=75	0,7995; k=69	0,7995; k=73
M=3	0,8305; k=71	0,8292; k=67	0,8279; k=69	0,8279; k=63	0,8266; k=65

Taula 38. Taula de valors de k i Recall de l'assignatura de Mecànica, Df2

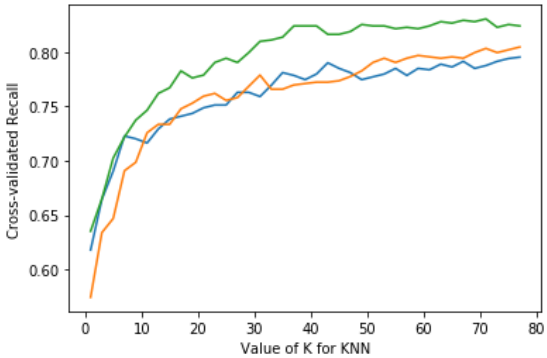


Figura 73. Gràfic paràmetre K-Recall de Mecànica, Df 2

- *Dataframe 3*

M=1	0,7808; k=41	0,7773; k=43	0,7761; k=39	0,7738; k=45	0,7738; k=37
M=2	0,8053; k=81	0,8030; k=77	0,8007; k=79	0,7984; k=75	0,7949; k=73
M=3	0,7961; k=75	0,7926; k=79	0,7914; k=77	0,7879; k=81	0,7868; k=73

Taula 39. Taula de valors de k i Recall de l'assignatura de Mecànica, Df3

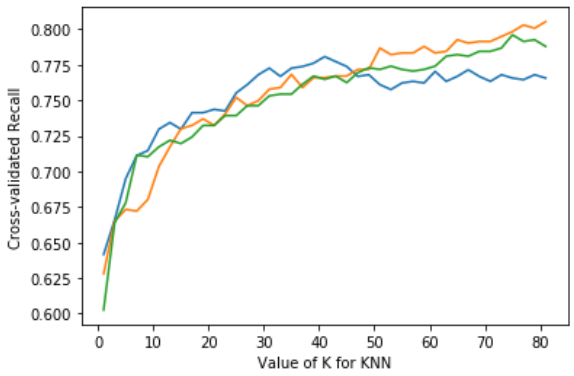


Figura 74. Gràfic paràmetre K-Recall de Mecànica, Df 3

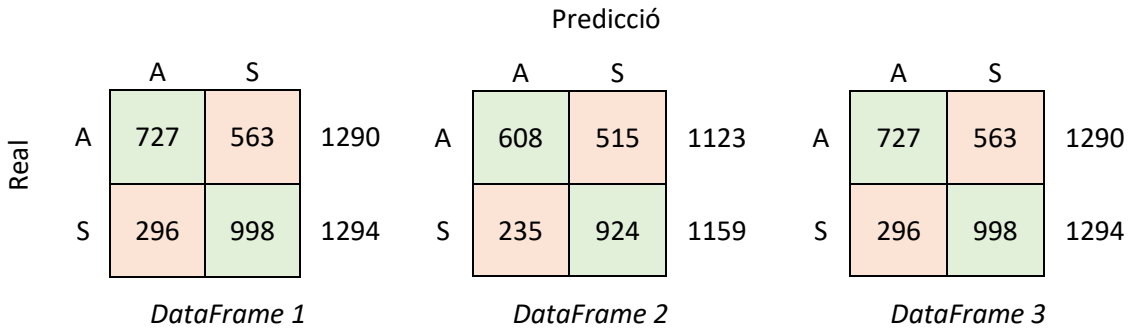


Figura 75. Matrius de confusió de l'assignatura de Mecànica

	k	CV-Recall	Accuracy	Precision	Recall	F1-Score
Dataframe 1	77	0,7978	0,6796	0,6486	0,7867	0,7105
Dataframe 2	77	0,8019	0,6902	0,6567	0,8179	0,7285
Dataframe 3	77	0,7972	0,6796	0,6486	0,7867	0,7105

Taula 40. Taula de valors de l'assignatura de Mecànica

El paràmetre *Recall* assoleix valors més elevats que el paràmetre *Precision.* El segon *DataFrame* és el que presenta valors dels paràmetres *Recall* i *F1-Score* superiors.

El paràmetre *Recall* obtingut mitjançant la validació interna és semblant al obtingut mitjançant la validació externa.

Conclusions

La diferència més destacada que s'observa és l'elecció del valor de k, el qual divideix les assignatures en dos grans grups. Per una banda es troba l'assignatura de *Mecànica* que pren valors elevats de l'híper-paràmetre k mentre que la resta d'assignatures adopta valors petits.

La principal causa d'aquest fet es troba en la distribució desequilibrada de dades. A excepció de *Mecànica,* la resta de les assignatures presenta aquest tipus de distribució, on la classe minoritària són els suspesos i la majoritària els aprovats.

L'algorisme de KNN classifica un nou punt de dades en funció de la majoria dels vots d'entre els seus veïns més propers k. Si es té una distribució de classes desequilibrada, la probabilitat que els veïns més propers d'aquest nou punt de dades pertanyin a la classe majoritària és més alta. El veí més proper del nou punt de dades pot pertànyer a la classe minoritària però si la resta de veïns (k-1) pertanyen a l'altre classe (per la seva major densitat a l'espai) el punt es classificarà erròniament. En aquest cas, la classificació d'un nou punt de dades a partir del seu veí més proper podria donar millors resultats, k=1.

Aquest fet es pot corroborar a partir dels resultats de les assignatures amb una distribució de dades desequilibrada, doncs el millor valor del paràmetre *Recall* correspon a una k=1. En altres paraules, a partir del veí més proper s'aconsegueix reduir el nombre de suspesos que es classifiquen com a aprovats. Com que k=1 no és un valor adequat a causa del *Overfitting,* s'opta per agafar el segon millor valor com k=3 o k=5 depenent del cas.

No obstant, classificar correctament la classe minoritària suposa no assolir tants bons resultats de la classe oposada. Per tal d'encertar el màxim nombre de suspesos, la validació creuada interna pren valors de k petits sense tenir en compte la predicció correcta dels aprovats. Tot i així, els valors de *Precision* són superiors als del paràmetre *Recall*. L'algorisme s'equivoca menys amb la predicció dels aprovats que amb la dels suspesos ja que és la classe majoritària.

Les assignatures de *Mètodes Numèrics, Equacions diferencials* i *Informàtica* presenten una distribució de dades altament desequilibrada amb només un 14-20% de suspesos aproximadament. El paràmetre de k pren valors petits i proporciona el màxim valor del paràmetre *Recall,* amb l'objectiu de minimitzar els suspesos classificats com a aprovats. Tot i així, el valor d'aquest paràmetre resulta ser molt baix juntament amb els paràmetres *Precision* i *F1 Score.*

Les assignatures d'*Electromagnetisme* i *Materials* presenten aproximadament un 30% de suspesos. La distribució de dades està menys desequilibrada que els casos anteriors amb un valor del paràmetre *Recall* més elevat.

Per les assignatures *d'Electromagnetisme, Materials* i *Informàtica,* a partir dels gràfics del valor de k en funció del paràmetre *Recall*, s'observa com les corbes de les seccions difereixen molt entre elles. Això és a causa de la variabilitat de les dades. Una possible solució seria dividir el conjunt de dades en més seccions.

Per últim, l'assignatura de *Mecànica* presenta una distribució de dades equilibrada amb aproximadament un 50% d'instàncies de cada classe. El valor de k creix a mesura que augmenta el valor del paràmetre *Recall* fins que s'estabilitza en un punt. Amb una distribució de dades equilibrada, no existeix cap classe predominant i es tria el paràmetre de k per tal que la predicció dels suspesos tingui la major quantitat d'encerts possible. Per el valor de k escollit, s'obtenen

valors de *Recall* superiors als valors de *Precision.* El valor del paràmetre *Accuracy* presenta un total d'encerts d'aproximadament el 68% respecte el total de les instàncies.

L'assignatura de *Mecànica* és la que classifica millor els suspesos a causa de les seva distribució de dades. Si els suspesos representen la classe minoritària, seran molt difícils de predir i no s'obtindran resultats desitjats. Una solució seria equilibrar les distribucions de dades per totes les assignatures per reduir les prediccions errònies dels suspesos i obtenir valors de k més elevats.

4.5. Comparació entre mètodes predictius

Es compararan com canvien els resultats quan s'aplica cadascun dels mètodes estudiats. Teòricament, KNN hauria de presentar millors valors en el paràmetre *Recall*, doncs s'ha escollit una k específica en cada cas per maximitzar aquest paràmetre.

Mètodes Numèrics, Equacions diferencials i Informàtica són les que presenten una distribució de dades més desequilibrada i conseqüentment les que obtenen pitjors resultats.

Mitjançant l'elecció de l'híper-paràmetre k en el mètode de KNN, s'observa una millora important del paràmetre *Recall* en les tres assignatures respecte la RL, sobretot en *Equacions diferencials* on arriba a duplicar el seu valor. L'augment d'aquests paràmetre s'assoleix sacrificant el paràmetre *Precision*, el qual es redueix. El valor *F1* també augmenta però ho fa en menor mesura, doncs no deixa de ser una mitjana dels altres dos paràmetres.

		Accuracy	Precision	Recall	F1-Score
LG	***Dataframe 1***	0,8630	0,7315	0,0439	0,0824
	Dataframe 2	0,8712	0,3333	0,0034	0,0067
	Dataframe 3	0,8603	0,6667	0,0137	0,0269
KNN	***Dataframe 1***	0,8231	0,2461	0,1208	0,1609
	Dataframe 2	0,8431	0,2369	0,0915	0,1303
	Dataframe 3	0,8460	0,3292	0,0879	0,1387

Taula 41. Resultats de l'assignatura de mètodes numèrics per a RL i KNN

		Accuracy	Precision	Recall	F1-Score
LG	***Dataframe 1***	0,8049	0,5004	0,0896	0,1494
	Dataframe 2	0,8098	0,6820	0,0383	0,0717
	Dataframe 3	0,8108	0,6505	0,0617	0,1108
KNN	***Dataframe 1***	0,7647	0,3301	0,2032	0,2514
	Dataframe 2	0,7774	0,3848	0,2314	0,2884
	Dataframe 3	0,7643	0,3290	0,2032	0,2511

Taula 42. Resultats de l'assignatura d'equacions diferencials per a RL i KNN

		Accuracy	Precision	Recall	F1-Score
LG	***Dataframe 1***	0,7988	0,5753	0,2187	0,3167
	Dataframe 2	0,8050	0,6617	0,1320	0,2152
	Dataframe 3	0,7988	0,5891	0,1934	0,2911
KNN	***Dataframe 1***	0,7535	0,3936	0,2892	0,3327
	Dataframe 2	0,7651	0,3985	0,2638	0,3171
	Dataframe 3	0,7535	0,3936	0,2892	0,3327

Taula 43. Resultats de l'assignatura d'Informàtica per a RL i KNN

Electromagnetisme i Materials: presenten una distribució de dades desequilibrada menys pronunciada que en el cas anterior.

L'aplicació de KNN per l'assignatura d'*Electromagnetisme* no presenta cap millora representativa respecte la regressió logística en ningun dels paràmetres. En general, el model de regressió logística presenta millors resultats que amb el model *K-Nearest Neighbors*.

L'assignatura de *Materials* presenta una millora en els resultats mitjançant l'aplicació de la tècnica KNN. El valor escollit de k ha permès augmentar el nombre d'encerts tant dels suspesos com dels aprovats.

		Accuracy	Precision	Recall	F1-Score
LG	***Dataframe 1***	0,7283	0,6256	0,4488	0,5213
	Dataframe 2	0,7235	0,6350	0,4185	0,5041
	Dataframe 3	0,7287	0,6427	0,4099	0,4988
KNN	***Dataframe 1***	0,6648	0,4895	0,4276	0,4555
	Dataframe 2	0,6722	0,5152	0,4315	0,4695
	Dataframe 3	0,6648	0,4895	0,4276	0,4555

Taula 44. Resultats de l'assignatura d'Electromagnetisme per a RL i KNN

		Accuracy	Precision	Recall	F1-Score
LG	***Dataframe 1***	0,7264	0,5884	0,3098	0,4055
	Dataframe 2	0,7165	0,6073	0,2627	0,3658
	Dataframe 3	0,7276	0,6084	0,2738	0,3772
KNN	***Dataframe 1***	0,6660	0,4320	0,3470	0,3849
	Dataframe 2	0,6661	0,4570	0,3638	0,4046
	Dataframe 3	0,6656	0,4312	0,3470	0,3846

Taula 45. Resultats de l'assignatura de Materials per a RL i KNN

Mecànica

Mecànica és l'assignatura que assoleix els millors valors de paràmetres en qualsevol dels mètodes predictius . Com s'ha comentat anteriorment, la seva distribució balancejada de dades permet arribar a aquests valors tan òptims. Mitjançant la tria del valor k es poden millorar els resultats.

		Accuracy	Precision	Recall	F1-Score
LG	***Dataframe 1***	0,6822	0,6662	0,7326	0,6975
	Dataframe 2	0,6880	0,6729	0,7515	0,7100
	Dataframe 3	0,6796	0,6625	0,7334	0,6957
KNN	***Dataframe 1***	0,6796	0,6486	0,7867	0,7105
	Dataframe 2	0,6902	0,6567	0,8179	0,7285
	Dataframe 3	0,6796	0,6486	0,7867	0,7105

Taula 46. Resultats de l'assignatura de Mecànica per a RL i KN

Conclusions

En general i prenent com a objectiu principal la predicció dels suspesos, es pot concloure que l'algorisme de KNN és millor que la regressió logística per les assignatures del Q3.

A excepció de l'assignatura d'*Electromagnetisme,* la resta d'assignatures milloren el seu paràmetre *Recall* amb l'aplicació de *K-Nearest Neighbors.* Aquest augment significa que el model prediu correctament més suspesos que la regressió logística. Tot i aquesta millora, els nous valors del paràmetre segueixen sent poc elevats. El millor valor correspon a l'assignatura de mecànica amb aproximadament un 80% de suspesos encertats respecte el total de casos predits. La resta d'assignatures presenten valors de *Recall* d'aproximadament 0,35 per materials, 0,27 per informàtica seguit d'un valor de 0,2 per equacions diferencials i 0,1 per mètodes numèrics. Com s'ha comentat anteriorment, aquests valors mínims es deuen al desequilibri entre classes.

L'assignatura d'*Electromagnetisme* no presenta una millora en la predicció dels suspesos quan s'aplica l'algorisme de KNN. És més, tant el paràmetre *Recall* com *Precision* disminueixen el seu valor. Per a aquesta assignatura, la regressió logística funciona millor com a predictor de suspesos i aprovats.

Quant als conjunts de dades, el primer *Dataframe* és el millor per el model de regressió logística. Aquest proporciona valors dels paràmetres *Recall* i *F1* més elevats a excepció de mecànica. Per a aquesta assignatura el segon *Dataframe* ha estat el millor.

En general, per al model de KNN i en base als paràmetres *Recall* i *F1,* el millor *Dataframe* és el segon. Es podria dir que aquest conjunt de dades alternatiu a l'original aporta més informació i proporciona una predicció més encertada sobre els suspesos. Per les assignatures de mètodes numèrics i informàtica, el primer conjunt de dades original segueix sent el que aporta millors resultats.

Es conclou que l'addició de la variable del nombre de repeticions per cada assignatura del tercer *Dataframe* en aquests models de predicció no aporta millores significatives.

5. PRESSUPOST

Els costos a considerar per realització d'aquest projecte es poden desglossar en costs de personal i costs d'infraestructura. Els costs descrits es troben representats a la *Taula 47.*

Costs de personal

Es calculen les hores dedicades a la realització del projecte i el cost del treball. El treball realitzat es pot dividir en cinc parts principals: introducció a la mineria de dades, comprensió del problema, comprensió i preparació de les dades, modelatge i validació. També es tindrà en compte les hores destinades a la redacció de la memòria

Costs d'infraestructura

Els costs d'infraestructura es componen de recursos informàtics, així com llicències i ordinador, i material d'oficina.

El software utilitzat per les etapes del procés de mineria de dades és de programari lliure i de codi obert, totalment gratuït. Les despeses en recursos informàtics corresponen a la utilització d'un ordinador i el preu de la llicència de Microsoft Office, tot i que es podria haver escollit una alternativa de programari lliure.

S'ha emprat un ordinador portàtil valorat en 850€ i es considera un cost de manteniment del 10% anual del seu preu d'adquisició per un ús de 1100 hores anuals. Sabent que l'ús del ordinador durant la realització del projecte és de 470 hores:

$$470\ h \cdot \frac{850\ € \cdot 0{,}1}{1100\ h} = 36{,}32€$$

A més del cost de manteniment de l'ordinador, cal tenir en compte el càlcul de la seva amortització. Es considera un ús de 47 setmanes a l'any durant 3 anys des del moment de la seva compra. Per a la realització del treball s'ha utilitzat durant 23 setmanes, aproximadament 5 mesos. Descomptant un dia de descans a la setmana, el seu ús real és de 20 setmanes.

$$20\ set \cdot \frac{\frac{850\ €}{3\ anys}}{47\ set} = 120{,}57€$$

L'ús del ordinador comporta un cost total de:

$$36{,}32 + 120{,}57 = \mathbf{156{,}89}€$$

Els costs de material d'oficina són derivats de l'ús de recursos com paper, bolígraf i altres residus generats durant el projecte. Es considera un cost d'aproximadament **15€.**

COSTS DE PERSONAL			
Concepte	**Dedicació (h)**	**Preu (€/h)**	**Cost (€)**
Introducció a la mineria de dades	25	15	375
Comprensió del problema	10	15	150
Comprensió i preparació de les dades	160	30	4800
Modelatge	130	30	3900
Validació	100	30	3000
Redacció de la memòria	45	20	900
COSTS D'INFRAESTRUCTURA			
Concepte			**Cost (€)**
Microsoft Office 365			69
Ordinador			156,89
Material d'oficina			15
COST TOTAL			**13.365,89**

Taula 47. Desglossament de costs del projecte

6. IMPACTE AMBIENTAL

L'impacte ambiental produït per aquest projecte és mínim. Les tasques s'han dut a terme mitjançant un ordinador, sense generar cap tipus de residu. El residu en forma de paper del material d'oficina ja ha estat contemplat a l'apartat de *Pressupost*.

L'únic element que es pot tenir en compte és el consum d'energia elèctrica per l'alimentació de l'ordinador i les emissions de CO_2 derivades d'aquest.

Es calculen les emissions de CO_2 a partir del mix elèctric. *El mix elèctric és el valor que expressa les emissions de CO_2 associades a la generació de l'electricitat que es consumeix* **[13]**. Segons l'Oficina Catalana del Canvi Climàtic, el mix de la xarxa elèctrica peninsular de 2018 s'estima en 321 g CO_2/kWh.

Es considera un consum de l'ordinador de 470 hores durant la realització del projecte. Si el consum de l'ordinador és d'aproximadament 150W, el consum d'energia és de 70,5kWh amb una emissió de 22,63kg de CO_2.

El consum energètic per part de l'enllumenat o la climatització del lloc de treball no es té en compte perquè s'ha treballat majoritàriament aprofitant llum natural i perquè aquest consum també és necessari quan no s'està treballant.

7. CONLUSIONS

Es considera que s'han assolit tots els objectius marcats inicialment. S'ha seguit una metodologia CRISP DM adaptada a les característiques del treball i s'han diferenciat les diferents fases i les tasques a realitzar en cadascuna d'elles, garantint la possibilitat de ser replicada a partir de la present documentació.

Quant a l'anàlisi de resultats, s'ha estudiat els paràmetres com la precisió, *Recall, Precision* i el valor *F1* dels models de predicció de la regressió logística i *K-Nearest Neighbors.* Per aquest últim, s'ha optimitzat l'híper-paràmetre k en funció de mètrica *Recall* per tal de minimitzar el nombre de suspesos predits erròniament. S'ha pres com a paràmetre k, definit com el nombre de veïns més propers, el corresponent a la mètrica més elevada. D'aquesta manera s'obté la millor classificació de suspesos possible. És un bon procediment si els costs associats a prediccions errònies són elevats i es volen minimitzar aquestes prediccions. Tanmateix, la tria d'aquest paràmetre k és un procés molt manual que requereix de bastant treball per part de l'analista.

La principal conclusió ha estat que l'algorisme de KNN prediu millor els suspesos que la regressió logística. Tot i així, la classificació no ha obtingut gaires bons resultats. Les mètriques de predicció dels suspesos han sigut baixes pels dos mètodes de predicció, sobretot en la regressió logística. El motiu principal es deu a la distribució desequilibrada de classes la qual provoca que el model detecti millor la classe majoritària, els aprovats. Els suspesos, per altra banda, representen la classe minoritària que conté menys exemples per entrenar fent que el model falli més. Tot i aquest desequilibri, podria ser que aplicant un altre tipus de model s'obtinguessin millors resultats.

També s'han comparat els paràmetres de precisió, *Recall* etc. entre diferents conjunts de dades per veure si l'addició de noves variables suposava una millora en el rendiment del model. En general, la introducció de dades externes així com la nota de selectivitat o el nombre de repeticions de les assignatures no ha suposat cap millora per el model de regressió logística. En canvi, per KNN la nota de selectivitat afegida al segon *Dataframe* millora el rendiment del model.

Tot i no haver obtingut uns paràmetres de rendiment dels models elevats en la predicció, s'ha assolit l'objectiu principal: l'estudi del rendiment de les tècniques de mineria de dades en la predicció de l'aprovat o suspès de les assignatures del Q3.

Conclusions personals

A nivell personal, es vol destacar els coneixements assolits al llarg d'aquest projecte. En primer lloc, s'ha pres consciència de la importància de la mineria de dades, la gran varietat d'aplicacions i utilitats que presenta i la informació capaç d'extreure d'un conjunt de dades. Les nocions de programació de *Python* adquirides en les assignatures d'informàtica del grau s'han pogut aplicar sobre un cas pràctic. S'ha familiaritzat amb la llibreria *Pandas* i les seves funcions que faciliten la manipulació de dades.

Treball futur

Es plantegen alternatives i millores d'anàlisi que es poden dur a terme amb la mateixa metodologia.

Es podria equilibrar la distribució de classes de manera que hi hagués el mateix percentatge de suspesos i aprovats en totes les assignatures mitjançant tècniques de *resampling.* D'aquesta manera, es milloraria la predicció de suspesos.

També es podria dur a terme l'aplicació d'optimització de paràmetres per la tècnica de regressió logística per tal d'assolir resultats més òptims.

BIBLIOGRAFIA

[1] Olson, David L. *Advanced Data Mining Techniques*. Berlin, Heidelberg: Springer, 2008. ISBN 978-3-540-76917-0.

[2] Hastie, T., Tibshirani, R. i Friedman, J. *The elements of Statistical Learning.* Springer [https://web.stanford.edu/~hastie/Papers/ESLII.pdf]

[3] McKinney, W. *Python for Data Analysis.* O'Reilly Media, Inc

[4] Witten, I., Frank, E. i Hall, M., *Data Mining.* Morgan Kaufmann Publishers, 2011. ISBN 978-0-12-374856-0

[5] Diversos autors. *Machine learning Mastery* [https://machinelearningmastery.com]

[6] *Pandas. Python Data Analysis Library* [https://pandas.pydata.org/]

[7] *NumPy* [https://numpy.org/]

[8] *Scikit-learn. Machine learning in Python* [https://scikit-learn.org/stable/]

[9] Diversos autors. *KDNuggets*. [https://www.kdnuggets.com/]

[10] Diversos autors. *Stackoverflow* [https://stackoverflow.com/]

[11] Diversos autors. *Towards data science.* [https://towardsdatascience.com/]

[12] Diversos autors. *Medium* [https://medium.com/]

[12] Diversos autors. *Cambridge coding.* https://cambridgecoding.wordpress.com

[13] *El canvi climàtic.* [https://canviclimatic.gencat.cat/ca/actua/factors_demissio_associats_a_lenergia/]

[14] Hajizadeh, Z., Taheri, M., i Johromi, M. *Nearest Neighbor Classification with Locally Weighted Distance for Imbalanced Data.* International Journal of Computer and Communication Engineering, Vol. 3, No. 2, March 2014 [https://pdfs.semanticscholar.org/a78e/4ea1ee1728f6b8538dd35ef39f0ebd19d412.pdf]

ANNEX

A1. Preparació de dades

Dataframe 1

```
import pandas as pd
data=pd.read_excel('qfaseini.xlsx')
faseini=pd.DataFrame(data)
data2=pd.read_excel('qfasenoini.xlsx')
fasenoini=pd.DataFrame(data2)

#Fase inicial
df=faseini[faseini.CODI_PROGRAMA==752].copy()
df1=df.sort_values(['CURS','QUAD'])
df2= df1[df1.QUAD != 0].copy()
df2['Superaini']=df2.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['SUPERA'].transform('last')
df3=df2.drop(['CODI_PROGRAMA','CREDITS','CURS','QUAD','NOTA_PROF','NOTA_NUM_AVAL','GRUP_CL
ASSE','SUPERA'],axis=1).copy()
df4=df3.dropna().copy()
df4['Notadef']=df4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['NOTA_NUM_DEF'].transform('mean')
df5=df4.drop_duplicates(['CODI_EXPEDIENT','CODI_UPC_UD','Notadef'],keep='last')
df6=df5.drop('NOTA_NUM_DEF',axis=1).copy()
df7=df6[df6.Superaini=='S'].copy()
df8=df7.drop('Superaini',axis=1).copy()
df8.reset_index(drop=True,inplace=True)
df9=df8.pivot_table(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Notadef')
df10=df9.dropna().copy()
#canviem nom a les assignatures
df10.rename(columns={240011: 'ALG', 240012: 'CALC1', 240013: 'MECFON', 240014: 'QUIM1', 240015:
'FONINFO', 240021: 'GEO', 240022: 'CALC2', 240023: 'TERMO', 240024: 'QUIM2', 240025: 'EXPRE'},
inplace=True)

#afegim assignatures primer quatri de segon
dfn=fasenoini[fasenoini.CODI_PROGRAMA==752].copy()
dfn1=dfn.sort_values(['CURS','QUAD'])
dfn2= dfn1[dfn1.QUAD != 0].copy()
dfn3=dfn2.drop(['CODI_PROGRAMA','CREDITS','CURS','QUAD','NOTA_PROF','NOTA_NUM_AVAL','GRUP_
CLASSE','SUPERA'],axis=1).copy()
dfn4=dfn3.dropna().copy()
dfn4['Notadef']=dfn4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['NOTA_NUM_DEF'].transform('first'
)
dfn5=dfn4.drop_duplicates(['CODI_EXPEDIENT','CODI_UPC_UD','Notadef'],keep='first')
dfn6=dfn5.drop('NOTA_NUM_DEF',axis=1).copy()
dfn6.reset_index(drop=True,inplace=True)
assignatures=['240132', '240133', '240131', '240033', '240031', '240032']
dfn7=dfn6.loc[dfn6['CODI_UPC_UD'].isin(assignatures)].copy()
dfn7.reset_index(drop=True, inplace=True)
#canviar els numeros a aprovats/suspesos
def nota(nota):
  if nota>=5:
    return 'A'
  else:
    return 'S'

dfn7['Notadef'] = dfn7['Notadef'].apply(lambda x: nota(x))
#Pivot without aggregation that can handle non-numeric data
dfn8=dfn7.pivot(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Notadef')
```

```
dfn8.rename(columns={'240132':'INFO','240133':'MEC','240131':'EDOS','240033':'MATERS','240031':'ELE
CTRO','240032':'METNUM'}, inplace=True)
dfn9=dfn8.dropna().copy()

#juntem
datadef=pd.merge(df10,dfn9,on='CODI_EXPEDIENT',how='inner')
datadef.to_pickle('datadef')
```

Dataframe 2

```
import pandas as pd

data=pd.read_excel('qfaseini.xlsx')
faseini=pd.DataFrame(data)

data2=pd.read_excel('qfasenoini.xlsx')
fasenoini=pd.DataFrame(data2)

data3=pd.read_excel('dadespersnomespreins.xlsx')
sele=pd.DataFrame(data3)

sele1=sele.drop(['SEXE','CP_FAMILIAR','ANY_ACCES','TIPUS_ACCES','CP_CENTRE_SEC'],axis=1).copy()
sele1['CODI_UPC_UD']='Sele'
sele2=sele1.rename(columns={'NOTA_ACCES': 'Notadef'})

#Fase inicial
df=faseini[faseini.CODI_PROGRAMA==752].copy()
df1=df.sort_values(['CURS','QUAD'])
df2= df1[df1.QUAD != 0].copy()
df2['Superaini']=df2.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['SUPERA'].transform('last')
df3=df2.drop(['CODI_PROGRAMA','CREDITS','CURS','QUAD','NOTA_PROF','NOTA_NUM_AVAL','GRUP_CL
ASSE','SUPERA'],axis=1).copy()
df4=df3.dropna().copy()
df4['Notadef']=df4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['NOTA_NUM_DEF'].transform('mean')
df5=df4.drop_duplicates(['CODI_EXPEDIENT','CODI_UPC_UD','Notadef'],keep='last')
df6=df5.drop('NOTA_NUM_DEF',axis=1).copy()
df7=df6[df6.Superaini=='S'].copy()
df8=df7.drop('Superaini',axis=1).copy()
df8.reset_index(drop=True,inplace=True)

df9=df8.append(sele2,ignore_index=True, sort=False)

#Normalitzem les dades
dt=df9['Notadef']
x=dt.values
x1=x.reshape(-1, 1)

from sklearn import preprocessing
# Create a minimum and maximum processor object
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x1)
dff = pd.DataFrame(x_scaled)

datanorm=pd.merge(dff,df9,right_index=True,left_index=True,how='inner')
datanorm1=datanorm.drop(['Notadef'],axis=1).copy()
datanorm1.rename(columns={0: 'Notadef'}, inplace=True)
```

```
df10=datanorm1.pivot_table(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Notadef')
df11=df10.dropna().copy()
#canviem nom a les assignatures
df11.rename(columns={240011: 'ALG', 240012: 'CALC1', 240013: 'MECFON', 240014: 'QUIM1', 240015:
'FONINFO', 240021: 'GEO', 240022: 'CALC2', 240023: 'TERMO', 240024: 'QUIM2', 240025: 'EXPRE'},
inplace=True)

#afegim assignatures primer quatri de segon
dfn=fasenoini[fasenoini.CODI_PROGRAMA==752].copy()
dfn1=dfn.sort_values(['CURS','QUAD'])
dfn2= dfn1[dfn1.QUAD != 0].copy()
dfn3=dfn2.drop(['CODI_PROGRAMA','CREDITS','CURS','QUAD','NOTA_PROF','NOTA_NUM_AVAL','GRUP_
CLASSE','SUPERA'],axis=1).copy()
dfn4=dfn3.dropna().copy()
dfn4['Notadef']=dfn4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['NOTA_NUM_DEF'].transform('first'
)
dfn5=dfn4.drop_duplicates(['CODI_EXPEDIENT','CODI_UPC_UD','Notadef'],keep='first')
dfn6=dfn5.drop('NOTA_NUM_DEF',axis=1).copy()
dfn6.reset_index(drop=True,inplace=True)
assignatures=['240132', '240133', '240131', '240033', '240031', '240032']
dfn7=dfn6.loc[dfn6['CODI_UPC_UD'].isin(assignatures)].copy()
dfn7.reset_index(drop=True, inplace=True)
#canviar els numeros a aprovats/suspesos
def nota(nota):
   if nota>=5:
      return 'A'
   else:
      return 'S'

dfn7['Notadef'] = dfn7['Notadef'].apply(lambda x: nota(x))
#Pivot without aggregation that can handle non-numeric data
dfn8=dfn7.pivot(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Notadef')
dfn8.rename(columns={'240132':'INFO','240133':'MEC','240131':'EDOS','240033':'MATERS','240031':'ELE
CTRO','240032':'METNUM'}, inplace=True)
dfn9=dfn8.dropna().copy()

#juntem
datadef=pd.merge(df11,dfn9,on='CODI_EXPEDIENT',how='inner')
datadef.to_pickle('datadef2')
```

Dataframe 3

```
import pandas as pd
data=pd.read_excel('qfaseini.xlsx')
faseini=pd.DataFrame(data)

data2=pd.read_excel('qfasenoini.xlsx')
fasenoini=pd.DataFrame(data2)

#Fase inicial
df=faseini[faseini.CODI_PROGRAMA==752].copy()
df1=df.sort_values(['CURS','QUAD'])
df2= df1[df1.QUAD != 0].copy()
df2['Superaini']=df2.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['SUPERA'].transform('last')
```

```
df3=df2.drop(['CODI_PROGRAMA','CREDITS','CURS','QUAD','NOTA_PROF','NOTA_NUM_AVAL','GRUP_CL
ASSE','SUPERA'],axis=1).copy()
df4=df3.dropna().copy()
df4['Notadef']=df4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['NOTA_NUM_DEF'].transform('mean')
df4['Conv']=df4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD']).cumcount()
df5=df4.drop_duplicates(['CODI_EXPEDIENT','CODI_UPC_UD','Notadef'],keep='last')
df6=df5.drop('NOTA_NUM_DEF',axis=1).copy()
df7=df6[df6.Superaini=='S'].copy()
df8=df7.drop('Superaini',axis=1).copy()
df8.reset_index(drop=True,inplace=True)

#Normalitzem les dades
dt=df8[['Notadef','Conv']]
x=dt.values
from sklearn import preprocessing
# Create a minimum and maximum processor object
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
dff = pd.DataFrame(x_scaled)

datanorm=pd.merge(dff,df8,right_index=True,left_index=True,how='inner')
datanorm1=datanorm.drop(['Notadef','Conv'],axis=1).copy()
datanorm1.rename(columns={0: 'Notadef', 1: 'Conv'}, inplace=True)


#taula 1
df9=datanorm1.pivot_table(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Notadef')
df10=df9.dropna().copy()
#canviem nom a les assignatures
df10.rename(columns={240011: 'ALG', 240012: 'CALC1', 240013: 'MECFON', 240014: 'QUIM1', 240015:
'FONINFO', 240021: 'GEO', 240022: 'CALC2', 240023: 'TERMO', 240024: 'QUIM2', 240025: 'EXPRE'},
inplace=True)

#taula 2
df11=datanorm1.pivot_table(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Conv')
df12=df11.dropna().copy()
#canviem nom Convocatories (n. repeticions)
df12.rename(columns={240011: 'C-ALG', 240012: 'C-CALC1', 240013: 'C-MECFON', 240014: 'C-QUIM1',
240015: 'C-FONINFO', 240021: 'C-GEO', 240022: 'C-CALC2', 240023: 'C-TERMO', 240024: 'C-QUIM2',
240025: 'C-EXPRE'}, inplace=True)

#unim taules
df13=pd.merge(df10,df12,on='CODI_EXPEDIENT',how='inner')

#afegim assignatures primer quatri de segon
dfn=fasenoini[fasenoini.CODI_PROGRAMA==752].copy()
dfn1=dfn.sort_values(['CURS','QUAD'])
dfn2= dfn1[dfn1.QUAD != 0].copy()
dfn3=dfn2.drop(['CODI_PROGRAMA','CREDITS','CURS','QUAD','NOTA_PROF','NOTA_NUM_AVAL','GRUP_
CLASSE','SUPERA'],axis=1).copy()
dfn4=dfn3.dropna().copy()
dfn4['Notadef']=dfn4.groupby(['CODI_EXPEDIENT','CODI_UPC_UD'])['NOTA_NUM_DEF'].transform('first'
)
dfn5=dfn4.drop_duplicates(['CODI_EXPEDIENT','CODI_UPC_UD','Notadef'],keep='first')
dfn6=dfn5.drop('NOTA_NUM_DEF',axis=1).copy()
dfn6.reset_index(drop=True,inplace=True)
assignatures=['240132', '240133', '240131', '240033', '240031', '240032']
```

```
dfn7=dfn6.loc[dfn6['CODI_UPC_UD'].isin(assignatures)].copy()
dfn7.reset_index(drop=True, inplace=True)
#canviar els numeros a aprovats/suspesos
def nota(nota):
  if nota>=5:
    return 'A'
  else:
    return 'S'

dfn7['Notadef'] = dfn7['Notadef'].apply(lambda x: nota(x))
#Pivot without aggregation that can handle non-numeric data
dfn8=dfn7.pivot(index='CODI_EXPEDIENT', columns='CODI_UPC_UD', values='Notadef')
dfn8.rename(columns={'240132':'INFO','240133':'MEC','240131':'EDOS','240033':'MATERS','240031':'ELE
CTRO','240032':'METNUM'}, inplace=True)
dfn9=dfn8.dropna().copy()

#juntem
datadef=pd.merge(df13,dfn9,on='CODI_EXPEDIENT',how='inner')
datadef.to_pickle('datadef3')
```

A2. Modelatge i validació

K-Nearest Neighbors

1.Selecció del híper-paràmetre

```
import pandas as pd
import numpy as np
import math
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import StratifiedKFold
import matplotlib.pyplot as plt

#datadef1,datadef2 o datadef3
df = pd.read_pickle('datadef')
df.reset_index(drop=True,inplace=True)
# create design matrix X and target vector y
X=df[['ALG','CALC1','MECFON','QUIM1','FONINFO','GEO','CALC2','TERMO','QUIM2','EXPRE']]
#ELECTRO,INFO,METNUM,MATERS,EDOS,MEC
y=df['MEC']

#Cerca del hiperparametre
print(df.shape)
klim=int(math.sqrt(2584*(2/3))*2)
print('Klim=',klim)

from sklearn.model_selection import GridSearchCV

outer_cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=1)
inner_cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=1)

matriu=[]
# outer folds
for i, (train_idx, test_idx) in enumerate(outer_cv.split(X, y)):
  print('\n[Outer fold %d/3]' % (i + 1))
  #Training and testing set of the outer fold
```

```
    X_train, X_test = X.loc[train_idx], X.loc[test_idx]
    y_train, y_test = y.loc[train_idx], y.loc[test_idx]

    print(len(X_train.index))
    # hyperparameter tuning by grid search CV
    knn=KNeighborsClassifier()
    param_grid = {'n_neighbors':np.arange(1,klim,2)}
    gs = GridSearchCV( knn, param_grid, scoring=['recall','precision','f1'],refit='recall',cv=inner_cv)
    gs.fit(X_train, y_train)
    #print(gs.cv_results_)
    recalls=gs.cv_results_['mean_test_recall']

    ks=np.arange(1,klim,2)
    l=list(map(list,zip(recalls,ks)))
    #print(l)
    l.sort()
    l.reverse()
    matriu.append(l[:5])
    print('\nList of [recall,k]',matriu[i])
    print(gs.best_score_,gs.best_params_,gs.best_index_,gs.cv_results_['params'][gs.best_index_])
    plt.plot(ks,recalls)
    plt.xlabel('Value of K for KNN')
    plt.ylabel('Cross-validated Recall')

print('\nMatriu:(3Fold x5 KValues))\n' , np.array(matriu))
```

2.K-Cros Validation, k=3

```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score,cross_val_predict
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.model_selection import StratifiedKFold

#datadef1,datadef2 o datadef3
df = pd.read_pickle('datadef')
df.reset_index(drop=True,inplace=True)
# create design matrix X and target vector y
X=df[['ALG','CALC1','MECFON','QUIM1','FONINFO','GEO','CALC2','TERMO','QUIM2','EXPRE']]
#ELECTRO,INFO,METNUM,MATERS,EDOS,MEC
y=df['MEC']

#Es selecciona manualment el valor de K
knn_cv = KNeighborsClassifier(n_neighbors =1041)
kfold = StratifiedKFold(n_splits=3, shuffle=True, random_state=1)

#Accuracy
cv_scores = cross_val_score(knn_cv,X,y,cv=kfold,scoring='accuracy')
scores_mean=np.mean(cv_scores)
#print('\nScores:',cv_scores)
print('cv_scores mean:{}'.format(scores_mean))

#Precision
cv_precision = cross_val_score(knn_cv,X,y,cv=kfold,scoring='precision')
cv_precision_mean=np.mean(cv_precision)
```

```
#print('\nScores_Precision:',cv_precision)
print('cv_precision mean:{}'.format(cv_precision_mean))


#Recall
cv_recall = cross_val_score(knn_cv,X,y,cv=kfold,scoring='recall')
cv_recall_mean=np.mean(cv_recall)
#print('\nScores_Recall:',cv_recall)
print('cv_recall mean:{}'.format(cv_recall_mean))


#F1
cv_f1 = cross_val_score(knn_cv,X,y,cv=kfold,scoring='f1')
cv_f1_mean=np.mean(cv_f1)
#print('\nScores_f1:',cv_f1)
print('cv_F1 score mean:{}'.format(cv_f1_mean))

#Predicció de la variable Y del nou conjunt de variables X
y_pred = cross_val_predict(knn_cv, X, y, cv=5)

#Build a text report showing the main classification metrics
print('\nClassification Report:\n',classification_report(y, y_pred))

from sklearn.metrics import precision_recall_fscore_support as score
print('\nPrecision, Recall, F1-Score\n',score(y, y_pred, beta=1.0, labels=None, pos_label=1,
average='binary', warn_for=('precision', 'recall', 'f-score'), sample_weight=None))

#Matriu de confussió
conf_mat = confusion_matrix(y, y_pred)
print('\nTest Confusion Matrix:\n',conf_mat)

import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(9,9))
sns.heatmap(conf_mat, annot=True, fmt='.3f', linewidths=.5, square = True, cmap = 'Blues_r');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Accuracy Score: {0}'.format(scores_mean)
plt.title(all_sample_title, size = 15);
```

Regressió logística

```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix,classification_report,make_scorer, precision_score,
recall_score, f1_score
from sklearn.model_selection import StratifiedKFold

#datadef1,datadef2 o datadef3
df = pd.read_pickle('datadef3')
df.reset_index(drop=True,inplace=True)
# create design matrix X and target vector y
X=df[['ALG','CALC1','MECFON','QUIM1','FONINFO','GEO','CALC2','TERMO','QUIM2','EXPRE']]
#ELECTRO,INFO,METNUM,MATERS,EDOS,MEC
y=df['MEC']
print('\nDf Dimensions:',df.shape)
```

```
#Import the model you want to use
#Make an instance of the Model. All parameters not specified are set to their defaults
from sklearn.linear_model import LogisticRegression
logistic_regression= LogisticRegression(solver='lbfgs')
kfold = StratifiedKFold(n_splits=3, shuffle=True, random_state=1)

#Accuracy
from sklearn.model_selection import cross_val_score
cv_scores = cross_val_score(logistic_regression,X,y,cv=kfold,scoring='accuracy')
scores_mean=np.mean(cv_scores)
#print('\nScores:',cv_scores)
print('cv_scores mean:{}'.format(scores_mean))

#Precision
cv_precision = cross_val_score(logistic_regression,X,y,cv=kfold,scoring='precision')
cv_precision_mean=np.mean(cv_precision)
#print('\nScores_Precision:',cv_precision)
print('cv_precision mean:{}'.format(cv_precision_mean))


#Recall
cv_recall = cross_val_score(logistic_regression,X,y,cv=kfold,scoring='recall')
cv_recall_mean=np.mean(cv_recall)
#print('\nScores_Recall:',cv_recall)
print('cv_recall mean:{}'.format(cv_recall_mean))


#F1
cv_f1 = cross_val_score(logistic_regression,X,y,cv=kfold,scoring='f1')
cv_f1_mean=np.mean(cv_f1)
#print('\nScores_f1:',cv_f1)
print('cv_F1 score mean:{}'.format(cv_f1_mean))

#Predicció de la variable Y del nou conjunt de variables X
from sklearn.model_selection import cross_val_predict
y_pred = cross_val_predict(logistic_regression, X, y, cv=kfold)

#Build a text report showing the main classification metrics
print('\nClassification Report:\n',classification_report(y, y_pred))

from sklearn.metrics import precision_recall_fscore_support as score
print('\nPrecision, Recall, F1-Score\n',score(y, y_pred, beta=1.0, labels=None, pos_label=1,
average='binary', warn_for=('precision', 'recall', 'f-score'), sample_weight=None))

#Matriu de confussió
conf_mat = confusion_matrix(y, y_pred)
print('\nTest Confusion Matrix:\n',conf_mat)

import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(9,9))
sns.heatmap(conf_mat, annot=True, fmt='.3f', linewidths=.5, square = True, cmap = 'Blues_r');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Accuracy Score: {0}'.format(scores_mean)
plt.title(all_sample_title, size = 15);
```